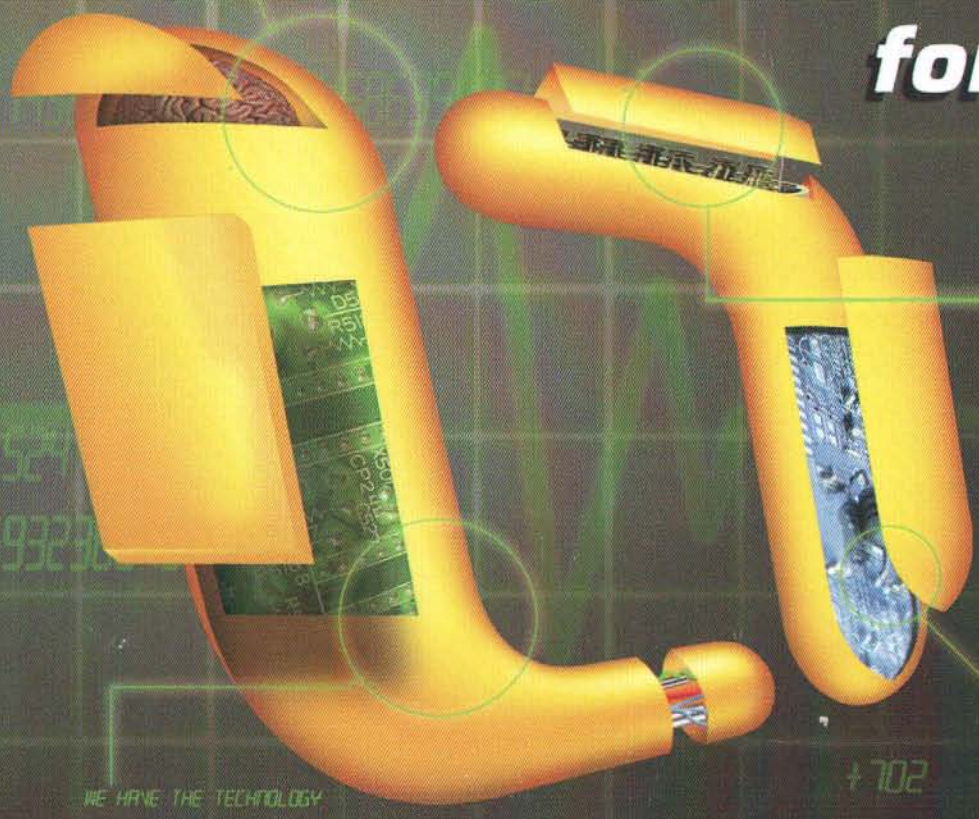
Using Git to Share Your Xcode Projects

MacTech Magazine
March • 2008

MACTECH®

The Journal of Macintosh Technology

Microsoft Office 2008 for Mac:



FASTER... SMARTER... BETTER?

Geoff Perlman on REALbasic 2008

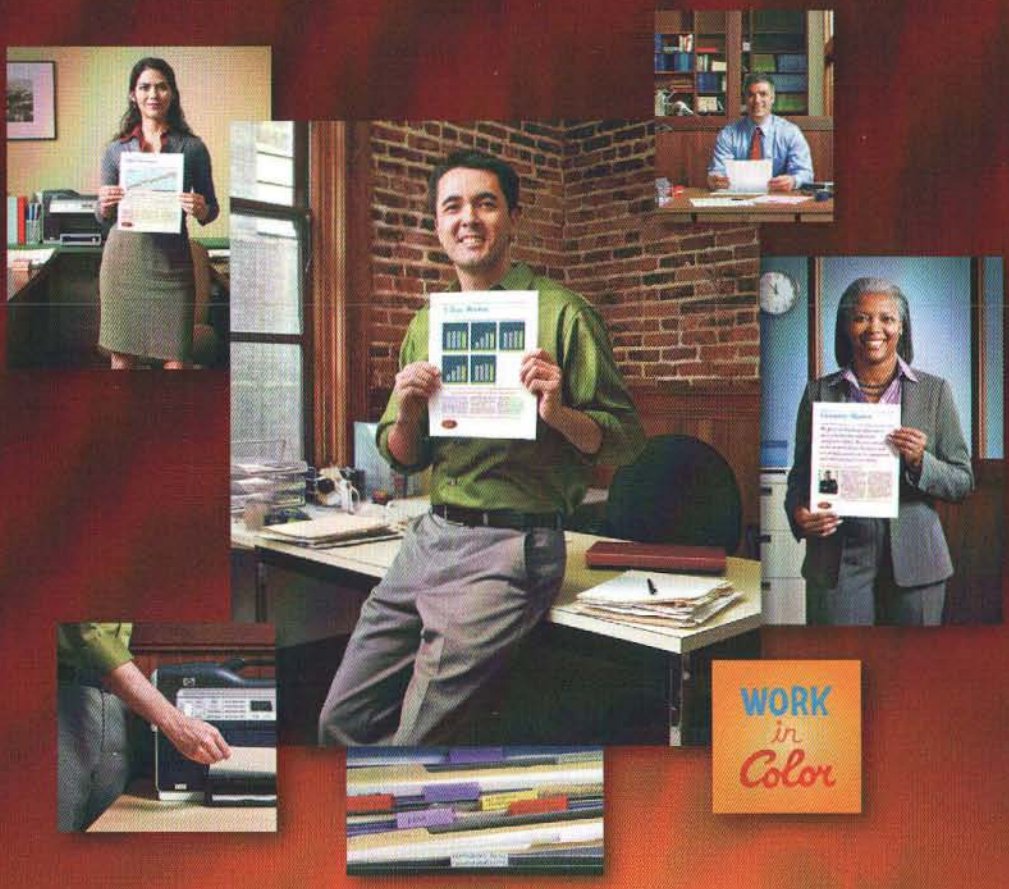
The Road to Certification: Pro Apps

What's in Your Target?

MACTECH.COM

$8.95 US, $12.95 Canada

ISSN 1067-8360 Printed in U.S.A.



UP TO 25% LOWER COST PER PAGE THAN LASER PRINTERS.*
HP OFFICEJET PRO ALL-IN-ONES STARTING AT $299.

They fax, scan, copy and deliver laser-quality printing—at an impressively low cost per page. Individual ink cartridges save money too, because you only replace what runs out. HP, *PC Magazine*'s Readers' Choice for service and reliability for 15 years running.

BEST
HP Officejet Pro L7780 AiO.
Plus wireless networking.
$499.

BETTER
HP Officejet Pro L7680 AiO.
Plus two-sided printing.
$399.

GOOD
HP Officejet Pro L7580 AiO.
Up to 35 pages a minute.
$299.

To learn more, visit hp.com/go/doesitall. Call 1-800-888-3119. Or visit your local reseller or retailer.

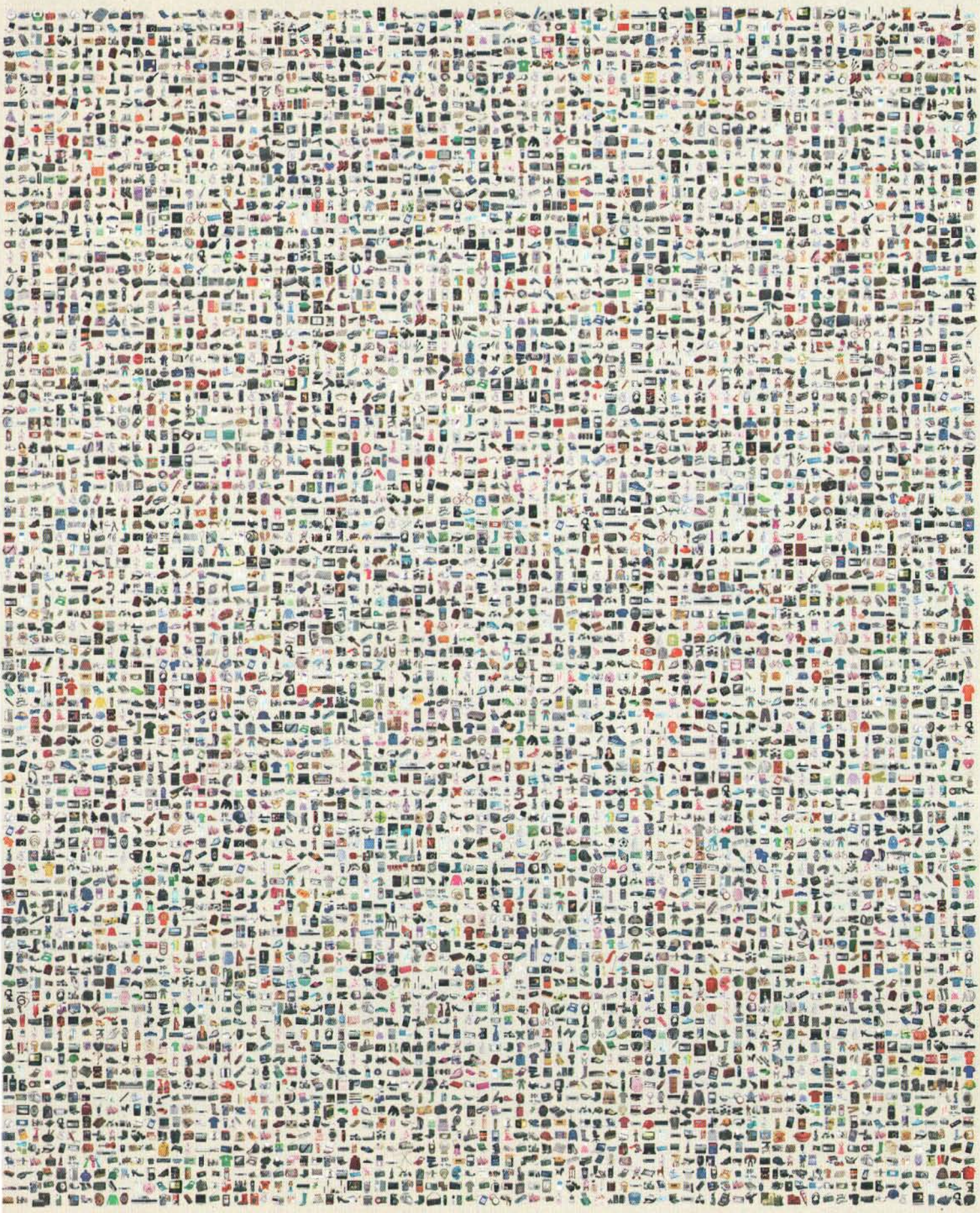
*Cost per page (CPP) laser supplies comparisons based on published manufacturer specifications of the highest capacity cartridges available for color laser all-in-ones under $1,000 and mono laser all-in-ones under $500, as reported by Current Analysis, Inc. as of July 2006. Officejet Pro L7500/L7600/L7700 AiO CPP based on HP 88 Large Ink Cartridges (not included, sold separately) estimated street price and published yield. Results may vary. Estimated U.S. retail price. Actual price may vary. Simulated images.
©2007 Hewlett-Packard Development Company, L.P.

THINGS MADE IN CHINA

COMPRESSED BY

STUFFIT DELUXE

SOFTWARE FOR MAC/PC



DOWNLOAD AT STUFFIT.COM/COPYRIGHT ® AND © 2006 ALLUME SYSTEMS, INC.

MacMall

MacMall is the #1 Apple Direct Reseller &

Introducing MacBook

The world's thinnest & most elegant notebook!

MacBook™ Air

- Up to 1.8GHz Intel Core 2 Duo processor ■ Only 0.76" thin & weighs just 3lbs.!
- 80GB hard drive or 64GB solid-state drive ■ 802.11n Wi-Fi wireless
- 13.3" glossy widescreen LED backlit display ■ Full-sized QWERTY keyboard
- Built-in iSight™ video conferencing ■ Up to 5 hours of battery life

starting at **$1794!*** #7373085
Finance for $54/mo.

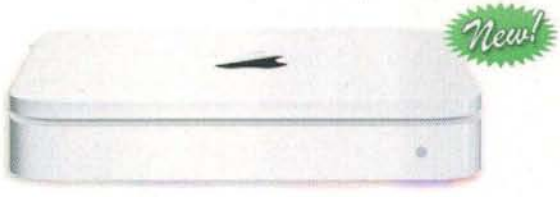
The amazing MacBook Air is thin enough to fit in a manila envelope! It's the ultimate blend of slipstream design and jet-powered productivity!

Only 0.76" thin!
Weighs just 3lbs.!
Up to 5 hours of battery life!

NEW Apple TV!

- Buy TV episodes and music videos from the iTunes Store and watch them on your standard or high-definition widescreen TV
- Rent movies online and watch them instantly
- Connects to your network at 802.11n wireless speeds or through an Ethernet connection
- Enjoy your computer's music collection, photo library and digital video on your home theater
- Access your .Mac Web Gallery or Flickr photo
- HDMI output ■ 1080p compatible

40GB only **$224!** #7185347



NEW Apple Time Capsule!

- Works with Time Machine in Mac OS X Leopard to back up everything on your system
- Automatic wireless or wired backup
- Functions as a full-featured AirPort® Extreme Base Station with 802.11n wireless
- Connect to multiple Macs and PCs
- Share a printer throughout your wireless network via the built-in USB 2.0 port
- 128-bit WEP encryption and NAT firewall
- Available in 500GB and 1TB models

500GB only **$294!** #7373106

VMware Fusion for Mac

#7288411

~~$67.99~~ **$47.99!**

After $20 mfr. mail-in rebate. Price before rebate is $67.99. See below.†

Parallels Desktop 3.0 for Mac

#7257816

FREE!

After rebate with Apple computer purchase. See below.†

Apple Final Cut® Studio 2

#7236215

upgrade **$448.99!**

NEW! Microsoft Office 2008!

New versions of your favorite office applications with support for both Intel and PowerPC Macs!

upgrade only **$209.94!** #7349124

*WE'LL BEAT ANY PRICE ON ANY APPLE BRAND PRODUCTS or competitive promotion on any Apple computer. We may require proof of dealer's pricing. Competitor must be an Apple Authorized Reseller. Product must be in stock at the competition. Advertised item must be identical to the one offered by MacMall and must be a new item in a factory sealed box. We reserve the right to limit quantity ordered. This offer does not apply to special, bonus or free offers, grand openings, special purchases or special buys, manufacturer's rebates, closeouts, clearances or "to dealers" one-of-a-kind or other limited quantity offers. Nor does it apply to advertising errors made by any authorized dealer. Not open to dealers. Void where prohibited. Sorry, offer not valid in conjunction with other rebate offers. On multiple-item quotes: "We'll Beat Any Price" applies to the price of the entire order, not individual line items. Offer applies to pre-taxed prices only. Offer valid only at time of purchase, not after the fact. ■ FREE SHIPPING OFFER-Free UPS ground shipping, after mail-in rebate, for most orders $99 or more (not including taxes and shipping costs). Limit one rebate per mailing address. Total rebate amount not to exceed $40 per shipping address. Valid only if the product(s) is purchased at the price listed on the Web on the day the order is placed. Rebate must be postmarked within 20 days of shipping (invoice) date. Offer good online or through MacMall's call center at 1-800-328-2794. Offer valid for all orders placed before

Authorized Reseller

the Mac pro's choice!

Air!

MacMall Exclusive Deals!
Get up to $200 cash back!*
FREE Parallels Desktop 3.0!*
FREE Epson Inkjet Printer!*
FREE Shipping!*
See below for details.
MacMall is your source for LCD and Plasma TVs!

NEW! Mac® Pro now with 8-core power!

- NEW! Up to 8-core performance with up to two Quad-Core Intel Xeon 5400 series processors at up to 3.2GHz
- NEW! ATI Radeon HD 2600 XT video graphics with 256MB GDDR3 memory
- NEW! PCI Express 2.0 graphics slot
- NEW! Four internal drive bays support up to 4TB of hard disk storage
- 16X SuperDrive™ with double-layer support (DVD±R DL/DVD±RW/CD-RW)
- Dual Gigabit Ethernet interfaces
- FireWire 800, FireWire 400 and USB 2.0

Mac Pro starting at
$2094!* *Finance for $63/mo.*

intel Xeon

1-800-328-2794 macmall.com Source Code: MACTECH

Apple MacBook™ Pro

- Intel Core 2 Duo processing at up to 2.6GHz
- NVIDIA GeForce 8600M GT professional graphics
- 15" models with mercury-free LED-backlit display
- 17" model with optional 1920 x 1200 display
- Built-in iSight™, wireless and Gigabit Ethernet

~~$1994~~ – $150 mail-in rebate
starting at **$1844!*** #7254324

Apple iMac®

- Intel Core 2 Duo processor at up to 2.4GHz or Intel Core 2 Extreme processor at 2.8GHz
- Up to 500GB HD ■ Glossy widescreen display
- ATI Radeon HD graphics ■ Aluminum casing
- Built-in iSight™, wireless and Gigabit Ethernet

~~$1194~~ – $50 mail-in rebate
starting at **$1144!*** #7288171

Your iPod Superstore!

With every flavor of player and the best special offers your can find, MacMall is your source for everything iPod! And you can visit us online for the widest selection of accessories at the best prices!

Call for iPods in all colors & capacities!

#7297622 iPod® shuffle™ (Blue) only $78.99
#7355255 iPod® nano (Silver) starting at $144
#7297631 iPod® classic (Silver) starting at $244

**Free case and shipping do not apply to iPod shuffle.

NEW iPod® touch!

- Includes NEW software revision 1.1.3 with Mail, Maps, Stocks, Weather and Notes
- Wi-Fi Web browsing ■ Multi-touch interface
- View movies, photos and YouTube videos
- Visit the iTunes Wi-Fi Music Store from any wireless hotspot to preview and download music

8GB only **$294!** #7373111

Adobe Creative Suite 3 Master Collection
Call for affordable upgrade paths!
#7228222

only **$2439.98!**

Apple Cinema Displays

#459836 20" Cinema Display® *was ~~$694~~ now* $594
#459837 23" Cinema HD Display *was ~~$994~~ now* $894
#459838 30" Cinema HD Display *was ~~$1934~~ now* $1794

Apple® AirPort® Extreme Base Station
802.11a/b/g/n
#7288377

only **$169.99!**

Mac OS® X Leopard™
#7355254

only **$109.98!**
#7355255 Family 5-Pack $179.99

11:59 p.m. PST on 2/29/08. Applies to UPS ground orders only ■ UP TO $200 CASH BACK OFFER-Get up to $200 Cash Back via MacMall mail-in rebate with purchase of select computer models. Ends 2/29/08. ■ FREE PRINTER OFFER-Get an Epson Stylus C120 Color Inkjet Printer FREE after $70 and $10 mfr. mail-in rebates with purchase of any Apple computer. Price before rebates is $60. Ends 3/31/08. ■ FUSION OFFER-Price after $20 mfr. mail-in rebate with purchase of any Apple CPU or Mac OS X Leopard. Price before rebate is $67.99. Offer expires 2/29/08. ■ FREE PARALLELS DESKTOP 3.0 OFFER-Get Parallels Desktop 3.0 for Mac (#7257816) FREE after $20 mfr. and $60 MacMall mail-in rebates with purchase of any new Intel-based Apple CPU. Price before rebates is $80. Ends 2/29/08. ■ FREE CASE OFFER-After $5.99 MacMall mail-in rebate with purchase of any iPod touch, iPod classic or iPod nano. Price before rebate is $5.99. Ends 2/29/08. ■ FREE ENGRAVING OFFER-Get select touch, iPod classic, iPod nano and iPod shuffle models engraved for FREE when you purchase them through MacMall. No rebate required. There is a $9.99 charge for select iPods. Call or visit www.macmall.com/myipod to place your order. ■ ALL OFFERS VALID ONLY WHILE SUPPLIES LAST. ■ †For rebate terms and conditions, please visit our Web site and enter the applicable part number. Download rebate coupons at www.macmall.com/rebates.

Apple Design Award
Best Mac OS X Leopard Application
2007 Runner-Up

You'll never go back to Quicken

iBank™

Your finances never looked so good.

Manage
- Bank Accounts
- Credit Cards
- Investments

Analyze
- Income
- Expenses

Plan
- Budgets
- Forecasts
- Schedule

Now available at these fine retailers:

MICRO CENTER The Ultimate Computer Store

Apple Store

TEKSERVE

amazon.com and you're done.

Office DEPOT

J&R JR.com 800 221 8180

IGG SOFTWARE

OfficeMax

Fry's ELECTRONICS

TABLE OF CONTENTS

ARTICLES & DEPARTMENTS

Mac in the Shell

Scripting with PHP

Forget that web stuff, PHP is a great scripting language

by Edward Marczak 8

The Road To Certification-Part 4: Pro Apps

Increase your knowledge and build credibility along the way

by Doug Hanley 16

Sharing with Git

Learn how to use Git to share your Xcode projects

by José R.C. Cruz 20

"Image"ine That!

A fresh look at creating deployment images

by Philip Rinehart, Yale University 34

The Road to Code

Building on a Solid Foundation

Exploring the Foundation framework

by Dave Dribin 36

Office 2008 Benchmarks

How well does Office 2008 run compared to Office 2004?

by Neil Ticktin 48

What's in Your Target

Unit testing and analysis coverage

by Aaron Montgomery; with Dave Dribin, contributing editor 58

Geoff Perlman on REALbasic

Catching up with Geoff and REALbasic 2008

by Norman Palardy 72

MacTech Spotlight

Austin Meyer

President, Laminar Research 80

From the Editor

At MacTech, we're busting at the seams with happiness because this issue is busting at the seams with content! We're fortunate to keep adding readers, and in the process we add contributors – people that want to share their knowledge about the Mac or some facet of the Mac. Of course, we're always looking for fresh voices and unique angles on this technology. If you're doing something cool with the Mac, let us know! Perhaps you – or someone you know – wants to write about it, or maybe you belong in the MacTech Spotlight. In either case, let us know! Send me an e-mail at exec-editor@mactech.com and let us know! First-time authors are always welcome.

Anytime that MacTech pits Mac against Mac for benchmarks, we end up with hugely popular articles. This month, we've run thousands of tests on G4 and Intel Macs running Tiger, Leopard, Office 2004 and the brand-new Office 2008. We'll show you where each is faster, slower and where you should care about that performance. Editor in Chief **Neil Ticktin** brings you the skinny on **Microsoft Office 2008 Benchmarks**.

We're closing in on the end of **Doug Hanley's Road to Certification** series. This time, Doug talks about getting certified in Pro Apps. Unless Apple changes something drastically, this is part 4 of this 5-part series on everything you could want to know about Apple certification tests.

José Cruz follows up last month's introduction to git with **Sharing With Git** – showing you how to use git for distributed projects *and* how to tie it all back in to Xcode. Git is gaining steam in the versioning system space, and we got the goods!

Dave Dribin continues down **The Road to Code** and delves into yet another critical piece to programming and understanding OS X – **the foundation framework**. Follow Dave's lead and learn how to write your *own apps*.

Returning author **Aaron Montgomery** brings us more goodies for developers and anyone who wants to understand **unit testing**. If you have a growing Xcode project, and you want to automate the testing of the code as it changes (and oh, remember to use a version system...something like git!), you should be reading, **"What's in Your Target?"**

Why Macworld is great: the power of bringing people together. **Norman Palardy** and I had the pleasure of sitting down with **Geoff Perlman** to talk about REAL Software and the latest release of REALbasic.

To round out this diverse and voluminous issue, our monthly features bring you **PHP as a general scripting language**, foundations of **creating system images**, and the **MacTech Spotlight** shines on **Austin Meyer**, best known for X-Plane. X-Plane is renowned for its realistic flight physics and the ability to fly vehicles from a two-seat prop plane to the Space Shuttle. Best part? All development is done on a Macintosh!

Don't forget: if you have a topic you'd like to share, let us know. If you're not up to covering a topic yourself, but know one that you want to learn about, let us know about that, too!

Thanks for reading, and we'll see you next month.

Ed Marczak,
Executive Editor

Communicate With Us

Department E-Mails

Orders, Circulation, & Customer Service
cust_service@mactech.com

Press Releases
press_releases@mactech.com

Ad Sales
adsales@mactech.com

Editorial
editorial@mactech.com
(Authors only, no pr)

Accounting
accounting@mactech.com

Marketing
marketing@mactech.com

General
info@mactech.com

Web Site
http://www.mactech.com

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail?**

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact MacTech Magazine Customer Service at 877-MACTECH

We love to hear from you! Please feel free to contact us with any suggestions or questions at any time.

Write to letters@mactech.com or editorial@mactech.com as appropriate.

MACTECH®
The Journal of Macintosh Technology

A publication of XPLAIN CORPORATION

The Magazine Staff

Publisher & Editor-in-Chief: Neil Ticktin
Executive Editor: Edward R. Marczak
Editor-at-Large: Dave Mark
Business Editor: Andrea Sniderman
Ad Director: Ken Spencer
Editor-at-Large, Open Source: Dean Shavit
Production: David Allen
Staff Writer: Jeffrey A. Rochin
Staff Writer: Marianne Shilpa Jacobie

Xplain Corporation Senior Staff

Chief Executive Officer: Neil Ticktin
President: Andrea J. Sniderman
Accounting: Marcie Moriarty
Customer Relations: Susan Pomrantz
Board of Advisors: Steven Geller, Alan Carsrud

Regular Columnists

Reviews/KoolTools: by Michael R. Harvey
The Source Hound: by Dean Shavit
Patch Panel: by John C. Welch
Mac In The Shell: by Ed Marczak

Board of Advisors

Chairman: Dave Mark,
Jordan Dea-Mattson, Steven Geller, Bruce Friedman, and Richard Kimes

Contributing Editors

Michael Brian Bentley, Gordon Garb, Vicki Brown, Chris Kilbourn
Marshall Clow, Rich Morin, Will Porter, Tom Djajadiningrat, Avi Rappoport,
Andrew S. Downs, Cal Simone, Steve Sisak

Canada Post: Publications Mail Agreement #41513541
Canada Returns to be sent to: Bleuchip International, P.O. Box 25542, London, ON N6C 6B2

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are $47.00 per year. Canadian subscriptions are $59.00 per year. All other international subscriptions are $97.00 per year. Domestic source code disk subscriptions are $77 per year. All international disk subscriptions are $97.00 a year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

All contents are Copyright 1984-2008 by Xplain Corporation. All rights reserved. MacTech and Developer Depot are registered trademarks of Xplain Corporation. RadGad, Useful Gifts and Gadgets, Xplain, DevDepot, Depot, The Depot, Depot Store, Video Depot, Movie Depot, Palm Depot, Game Depot, Flashlight Depot, Explain It, MacDev-1, THINK Reference, NetProfessional, NetProLive, JavaTech, WebTech, BeTech, LinuxTech, MacTech Central and the MacTutorMan are trademarks or service marks of Xplain Corporation. Sprocket is a registered trademark of eSprocket Corporation. Other trademarks and copyrights appearing in this printing or software remain the property of their respective holders.

by Edward Marczak

Scripting with PHP

Forget that web stuff, PHP is a great scripting language!

Introduction

Since the inception of this column, I've covered general shell tools, bash commands and general bash scripting. Life in the shell involves much more! There are many other scripting environments – and shells, for that matter! While I won't be stepping totally outside of bash, I will be looking at other ways to automate your environment. While I *like* bash, I don't necessarily *love* it. PHP – yes, the one you may know from web development – has a CLI component that makes a great scripting language. Of course, it's installed by default under OS X (version 5.2.4 as of OS 10.5.1). So let's dig in and learn some new techniques.

Why, oh Why?

Why am I forsaking bash? While bash is powerful, it falls down in some key areas, one of which being easy database access. Yes, using the tools we've talked about over the months, we could use the mysql binary, send output to standard out, grep-sed-and-awk our way to get what we're after. But that's not easy. Or elegant.

PHP is a dynamically and loosely typed language with extensions that allow easy access to data outside of its world. PHP supports access to raw network sockets, curl libraries allow access to URLs (ftp, http, https, etc.), and various database libraries allow access to various databases. This support needs to be compiled in, and Apple has made sure we have the tools we need. (You can look at everything that's compiled in by running `php -r 'phpinfo();' | less`. Once you're looking at that output, search for 'curl' and 'mysql').

PHP is also Open Source, like many of the packages that Apple includes with OS X. It was written by Rasmus Lerdorf in 1995 (as "PHP/FI" which evolved into the PHP we use today), and stands for "PHP: Hypertext Preprocessor". While its goal was for web development, it's also now nicely suited for scripting. (There wasn't always a CLI component).

What can PHP do? Just about anything! Well, anything that you'd use a scripting language for. PHP runs on just about every platform, and there is even a way to tie in a GUI (but that's beyond what I'm going to go into here. See http://gtk.php.net/ for more information).

Do the PHP

Nothing like an example to get things going. Most text editors recognize the .php filename extension and change modes accordingly. TextWrangler, BBEdit, vi and TextMate all have colorizations and in some cases, extra support for the PHP language and its constructs. So, fire up your favorite one, and type the venerable "hello world" program:

Listing 1: hello_world.php

```
#!/usr/bin/php
<?php
print "Hello, world!\n";
?>
```

As with all programs that we're going to run from the shell, we need to mark it executable if we intend to run it by name alone (chmod 770 hello_world.php). Notice that I used the shebang line here (line 1). Technically, you could omit that, and run the program like this:

```
php hello_world.php
```

However, I think the shebang line along with marking it executable underscores that we can treat PHP like a 'real' scripting language. This way, we can simply:

```
./hello_world.php
```

and be on our way. Let's look closer at listing 1.

First, we have the shebang line as discussed. Next, we have this odd-looking construct: "<?php". This simply signifies that what follows will be PHP code. As originally intended, PHP can be embedded in HTML documents. The PHP opening ("<?php") and closing ("?>") tags allow you to jump in and out of "PHP mode." (As an aside, if you're not in "PHP mode," you're in HTML mode, and the php engine will happily spit stray lines back to stdout. That's why there is no blank line in listing 1 between the shebang and the php opening tag. Also, the final closing tag, technically, is optional).

The next line contains a print statement, which simply outputs its arguments to standard out. In this case, the argument is the string "Hello, world!\n". The "\n" character is a newline, which `print` does not output on its own after printing.

Rekindle the LOVE for your computer.
"OWC, Thank You Very Much for Doing RAM Right"
- iBryan, The Digital Guy
FAST, HIGH QUALITY MEMORY
OWC
FAST, HIGH QUALITY MEMORY
OWC
PC5300 DDR2 667MHz
MacBook®, MacBook Pro, & iMac® Intel™ Memory Up to 4.0GB
We have the right memory for your Mac®!
www.owcmac.com/memory
Mac Pro Memory Up to 32GB
Other World Computing™
Visit: www.owcmac.com or Call: 800.275.4576
BizRate 2007
Other World Computing is a trademark, OWC and OWC logo are registered trademarks of Other World Computing. Other marks may be the trademark or registered trademark property of their owners. Prices, specifications, and availability are subject to change without notice.

Finally, we close with an ending php tag: "?>". All in all, you probably figured out this short program without explanation. One thing to note, though: unlike the bash constructs that we've seen, PHP expects each line to end with a semi-colon (";"). PHP uses this as a line separator, and the PHP engine will print errors if you omit it at the end of any line or between any separate statements.

Beyond The Beginning

Like any good programming environment, PHP supports comments, variables and all basic constructs such as flow control statements and loops. Let's get the easy one out of the way: good code is commented code.

PHP honors C, C++ and Perl style comments. Let's look at all three in action.

Listing 2: Comment example

```
<?php
/* example1.php
This short code snippet illustrates good code commenting.
Ed Marczak, 2008
*/

$numargs = func_num_args();       // This retrieves the
number of args passed
echo "Number of arguments: $numargs\n";
# Here's the closing tag:
?>
```

I'm sure I don't need to belabor that any further.

Variables in PHP are represented by a dollar sign followed by the name of the variable. The variable name *is* case-sensitive. A valid variable name starts with a letter or underscore (not a number) and is then followed by any number of letters, numbers or underscores. (With one exception: "$this" is a reserved variable name and cannot be overridden). As mentioned, PHP is both loosely and dynamically typed. This is a fancy way of saying variables take on the properties of the contents assigned to them, and generally 'do the right thing.' As always, an example:

Listing 3: PHP Variable Demo

```
<?php

$a = "apple";  // a is a string
if ($a=="orange") print "Orange\n";
$a=5        // a is now an integer
?>
```

(Note: the *final* closing tag can also act as a semi-colon, hence its omission in listing 3). In general, this aids the rapid development that can be done with PHP. However, if you ever do need to find out what type a variable is at the moment, you have several options. You can print it out using `getype()` or you can test for it with istype(). You can also use "===" in comparisons to ensure not only that variable contents are the same, but that types match also. You can force a variable to be a certain type by using settype(), or by *casting* the variable if you're familiar with that from other languages. I won't delve into this much deeper except to say to be careful with this. Again, the PHP engine will try to 'do the right thing.' Take listing 4 as your warning.

Listing 4: Casting example

```
<?php
$x=TRUE;   // x is a boolean
$x=(string)$x; // x now contains "1"
$x="9bar"; // string again
settype($x, integer); // x is now an int, and 0
?>
```

In a final show of the PHP engine trying to do the right thing, it will "juggle types" as needed. So, if $a is an integer and $b is a float, adding them together evaluates *everything* as a float, and returns a float. Strings also gain an implicit conversion if used with mathematical operators. "15" + 5 equals integer 20.

Speaking of types, PHP supports the following types:

Integer
Boolean
Float (aka "double" or "real")
String
Binary
Array
Object

We'll be exploring these types as we go.

Second Gear

Other things to note about how PHP deals with string variables and quoting. String literals can be specified as *single quoted*, *double quoted* or use *heredoc* syntax. I'll concentrate on the first two for now.

With single quotes, variables are *not* expanded, and only a backslash need be escaped. You can nest double quotes within single quotes. For example:

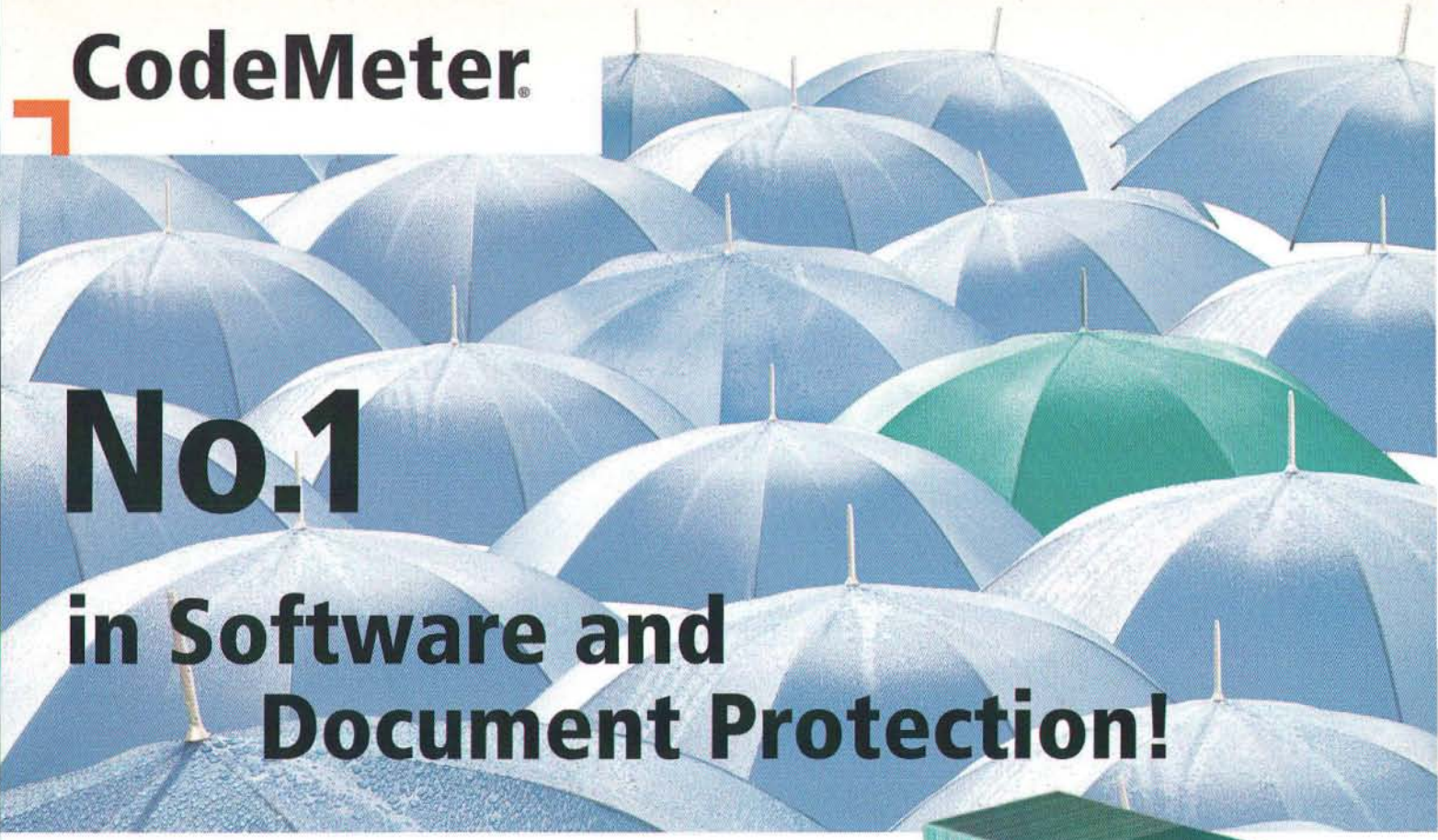
```
print 'Monty Python\'s Flying Circus.';
print 'I like $dollars';
print 'I\'m splitting
       this line';
```

The first line prints "Monty Python's Flying Circus." The second literally prints "I like $dollars." – without trying to evaluate "`$dollars`" as a variable. The third example shows that we can even embed the newline character into a single-quoted string.

If a string is enclosed in double quotes, PHP expands variables and interprets certain *escape sequences*. The major ones are:

\n	linefeed (0x0A, or 10 in ASCII)
\r	Carriage return (0x0D, or 13 in ASCII)
\t	Horizontal tab (HT or 0x09 (9) in ASCII)
\v	Vertical tab

CodeMeter®



No.1

in Software and Document Protection!

Highest Security

- Vendor selectable secret and private key.
- Strong encryption algorithms with AES 128-bit and ECC 224-bit.
- Best-in-class tools for automatic protection (envelope, without source modification) for Win32, Win64, .NET, Java and MacOS X Universal (PPC, Intel).

Best Flexibility

- More than 1000 independent licenses can be protected by one CM-Stick.
- One versatile hardware key for all license models including floating network licenses.
- Multi platform support including Windows, MacOS X and Linux.

New Distribution Channels

- License transfer by SOAP based CM-Talk or file based Field Activation Service in e-shops.
- Multiple-purpose, including protecting low cost software and digital content.

Unique End User Advantages

- First and smallest dongle with up to 2 Gbyte flash drive.
- No drivers necessary – can be used without administrator rights.
- CM Password Manager, secure virtual drive and secure login.

Order your Free Software Development Kit now!
Phone 1-800-6-GO-WIBU | order@wibu.us

2007 SIIA //CODiE// FINALIST
2005 SIIA //CODiE// FINALIST
2006 SIIA //CODiE// FINALIST

Mac

Universal

WIBU-SYSTEMS submitted the CodeMeter Password Manager and the CodeMeter SDK for the **Apple Design Awards 2007.**

WIBU SYSTEMS

WIBU®-SYSTEMS USA Inc.
110 W Dayton Street
Edmonds, WA 98020, USA
www.wibu.com
info@wibu.us
Tel: 1.800.6.GO.WIBU
1.425.775.6900
Fax: 1.206.237.2644

©2008 by WIBU-SYSTEMS USA, Inc. All rights reserved.

\f	Form feed (since PHP 5.2.5)
\\	Backslash
\$	Dollar sign
\"	Double quote

In action:

```
$num_dogs=6;
print "There are $num_dogs dogs\n";
```

This prints "There are 6 dogs" followed by a newline character.

Flow Control

All flow control deals with *comparison* for purposes of directing flow or knowing how many times to loop. PHP understands the following flow control comparison operators:

==	Equal
===	Identical
!=	Not equal
<>	Not equal
!==	Not identical
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Let's use these in a simple example:

Listing 5:Basic Flow Control

```
if ($a > $b) {
    print "a is greater than b\n";
    $top = $a;
} else {
    print "b is greater than a";
    $top = $b;
} .
```

PHP uses *curly brace* syntax to create a *statement group*. Listing 5 illustrates an `if` flow control structure. Generically, `if` tests an *expression* – any expression. If `a` is greater than `b`, the flow follows into the first group `if` statements. Otherwise, we run the statements in the `else` group.

Control structures can also be *nested*.

Listing 6: Nested control structures

```
$i=100;
while ($i<=500) {
    if (fmod($i,2)==0)
        print "$i\n";
    $i++;
}
```

You may note from listing 6 that if a control structure only has a single statement, curly braces can be omitted (though I recommend *always* retaining the braces).

Mimimax is the only compact keyboard with a true numeric keypad !
only 13" (33 cm) long!

16", 6 cm

For Mac OS X & Windows

Ideal for laptops !

After a surprisingly short adaptation time, you will type faster than ever, especially when one hand is tied up with the mouse, a document or telephone. Each version can write in the 11 major Western languages, including Spanish. Available in USB (with or without 2 USB 2.0 ports) and RF (with charging receiver).

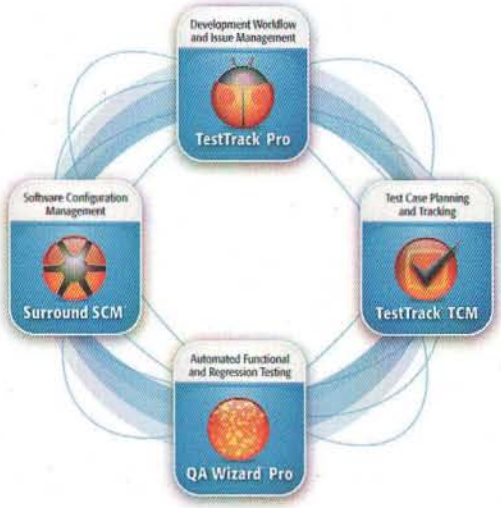
Please contact Jean-Michel KLEIN at +33 (0) 148.126.550 or jmklein@akor.fr.
http://www.akor.fr / AKOR 6, rue Rochebrune 93110 Rosny-sous-Bois (France)

Seapine Software

Satisfy Your Quality Obsession

Software quality and reliability are your lifelines to customer loyalty and profitability. Rigorous quality discipline and seamless collaboration between your development and QA teams will bring your customer-focused strategy to life.

With offices worldwide and over 8,500 customers, Seapine Software is the leading provider of integrated quality-centric application lifecycle management (ALM) solutions. For over 12 years Seapine's easy-to-use, award-winning tools have helped companies achieve success by streamlining communications, improving traceability, and making development and QA teams more productive.



Satisfy Your Quality Obsession.
Visit Seapine Software at **www.seapine.com/qualityready1**

Integrated ALM for a Quality-Ready Advantage

©2008 Seapine Software, Inc. Seapine, TestTrack Pro, TestTrack TCM, Surround SCM, QA Wizard Pro, and the Seapine logo are trademarks of Seapine Software, Inc. All Rights Reserved. All other trademarks are the property of their respective owners.

Shell Interaction

There are a few ways that you can have your PHP script interact with the shell in general. First, you can have PHP execute other shell commands. Second, you can pass arguments in to the script from the command line. Third, PHP is fully capable of reading from standard in and directing output to standard out and standard error.

The first one is pretty easy: to run a shell command – one that may have no built-in PHP equivalent – you simply use the backtick operator. The output of the shell command is assigned to the variable of your choice. Listing 7 shows this in action.

Listing 7: diskmon.php

```
#!/usr/bin/php
<?php

$freespace=trim(`df / | tail -1 | awk '{print $5}' | cut -
d "%" -f1`);

if ($freespace > 80) {
    print "System volume is at ${freespace}% full - you
may want to look at that.\n";
    die();
}

if ($freespace > 50) {
    print "System volume is at ${freespace}% full - seems
OK.\n";
    die();
}

if ($freespace >= 0) {
    print "System volume is only ${freespace}% full - no
problems.\n";
    die();
}
print "System volume is an indeterminate amount full.\n";
```

Accepting and handling command line arguments is also a straight-forward venture. PHP populates the variables `$argc` and `$argv` (an array) with the count of arguments and the contents of the arguments respectively. A simple example:

```
if ($argc < 3) {  // we need 2 actual arguments
    print "Usage: $argv[0] param1 param2";
    die();
}
print "You entered $argv[1] and $argv[2].\n";
```

The first element of $argv (which starts counting from zero) will contain the name of the program being run, as called from the command line. So, if someone symlinks to your program and it is called that way, $argv[0] will contain the name of the symlink.

Finally, PHP can easily handle something like this:

```
$ codeprep | php > accounting.csv
```

...where the codeprep application is actually outputting php code. Naturally, PHP will read standard in like so:

```
$ ls -l | list_filter.php
```

The program list_filter.php would contain a loop like this:

```
while ($line = trim(fgets(STDIN))) {
    // Process input here
    print "$line\n";
}
```

Standard out is standard out: all echo and print statements are sent there automatically. But what if you want to 'do the right thing' and send error output via standard error? Easy: just use fwrite to direct output to that stream:

```
fwrite (STDERR, "Record number $rec_no is malformed\n");
```

With this in your code, you can still do this:

```
$ data_gen.php > datafile.csv
Record number 70 is malformed
Record number 103 is malformed
```

You still end up with a good data file, but also can be alerted to exceptions.

Conclusion

PHP is just one of the many nice ways to get into, or continue scripting under OS X. The brilliant thing is that OS X treats all scripting languages pretty equally. Additionally, if you're already familiar with PHP from web development, it makes a nice and easy transition into scripting for the system environment. PHP-based scripts can be used for anything that bash or perl scripts are: triggered automation from cron, GUI interaction and even our beloved login hooks.

Next month, we'll dip further into PHP, interaction with MySQL and other PHP-based script topics.

Media of the month: The Illustrated World's Religions: A Guide to Our Wisdom Traditions by Huston Smith. This one has been around for a bit, but its easy reading overviews and beautiful photography make this a good general read and nice reference guide. Once you see the different perspectives in this book, perhaps we end the emacs / vi wars!

Until next month, keep scripting.

About The Author

Ed Marczak is the Executive Editor for MacTech Magazine, and has been lucky enough to have ridden the computing and technology wave from early on. From teletype computing to MVS to Netware to modern OS X, his interest was piqued. He has also been fortunate enough to come into contact with some of the best minds in the business. Ed spends his non-compute time with his wife and two daughters.

Record. Edit. Play.

WireTap Studio's revolutionary non-destructive audio editing technology lets you edit your recorded audio without any loss of quality or content.

Finally, an audio editor that's powerful, intuitive, and easy to use.

WireTap Studio

Download your free trial now: http://www.AmbrosiaSW.com/MacTech

AMBROSIA®
SOFTWARE INC

winner macworld eddys

The Mac Observer EDITORS CHOICE AWARD 2008

Audio Units · Mac · Universal · Works with .mac · WireTap Studio I'd Tap That!

Macworld 4½

WireTap Studio requires a Mac with an Intel, PowerPC G5, or PowerPC G4 (1GHz or faster) processor and Mac OS X 10.4 or later.
WireTap Studio, Ambrosia Software, Inc., and the Ambrosia Software logo are registered trademarks of Ambrosia Software, Inc. All other brand, product, service, and feature names or trademarks are the property of their respective owners.

The Road To Certification: Part 4: **Pro Apps**

Increase your knowledge and build credibility on the way

by Doug Hanley

Introduction

In this series of articles, we have looked at Apple's IT certifications and hardware certifications. We examined reasons for and benefits of getting certified, as well as the testing experience and the changes Apple made to its IT certifications with the release of Mac OS X Leopard, and their new Macintosh Technician certification, which qualifies a technician to perform warranty repairs while working at Apple Authorized Service Providers. In this article, we will review Apple's Pro Apps certifications for applications like Final Cut Pro and Logic Pro.

Apple offers a tiered certification program for Apple's professional digital applications. The supporting curriculum applies to creative professionals in varying capacities, including: editors, filmmakers, sound designers, photographers, special effects artists, teachers, and more.

Apple Certified Pro

Obtaining Pro Apps end-user certifications grants you the title of Apple Certified Pro. To become an Apple Certified Pro for an application, you need to pass the associated exam at an Apple Authorized Training Center (AATC). A Level One certification means that you have basic operational knowledge of the application, while a Level Two certification attests to a deeper understanding of the application. Level One exams are usually administered at the end of specific courses, but you can take Level One exams without taking a class, for a fee at an AATC that offers Pro Apps Classes. Level Two exams can be taken only in conjunction with the associated advanced classes and you need to pass the Level One exam before you can take the Level Two class and exam. The certifications are version-based for each application.

Once you are an Apple Certified Pro you can use the Apple Certified Pro logo on your business card and website. Certification gives you a way to differentiate yourself and validate your expertise in the digital media applications whose exams you pass. You may also join the Apple Certification Alliance (http://training.apple.com/alliance/) at no cost and have a free listing in the directory of Certified Pros on Apple's website.

You can obtain certification for the following applications: Final Cut Pro 6, Final Cut Express 4, Final Cut Studio 2 (Motion Graphics), DVD Studio Pro 4, Motion 3, Soundtrack Pro 2, Color, Aperture 1.5, Color Management, Logic Pro 7 and 8, Shake, and Xsan For Pro Video. Final Cut Pro and Logic Pro are the only two applications for which a Level Two certification is currently available.

The average class length is 3 days. There is a more extensive Final Cut Pro 6 class that is five days long. The Soundtrack Pro 2 course is only two days long. The courses do a great job of balancing concepts and theory with demos and hands-on exercises. The Apple Training Series books from Peachpit Publishing are used in each of the classes. If you want to become certified, the test is normally offered in the afternoon on the last day of class. You can find more information about the classes and course outlines at: http://www.apple.com/software/pro/training/courses/

Apple Certified Trainer Certification

The Apple Certified Trainer (ACT) program trains and tests qualified industry professionals to deliver Apple's certified curriculum. ACTs work with AATCs to teach

Upgrading to Leopard?

Don't forget to feed the beast.

Upgrade your system memory with Mushkin PerfectMatch to get the most out of OS-X Leopard

It's a jungle out there when it comes to operating system upgrades and the ever increasing minimum system requirements for optimal performance. Now, with Mushkin PerfectMatch memory, you won't have to play guessing games and choose between the increased functionality of Leopard and your current version of OS-X.

Because every PerfectMatch Apple memory module is built from components carefully selected to provide true compatibility and match or exceed the performance of the original Apple memory installed at the factory. Which means you can tame the beast and confidently and easily upgrade the performance of your system to run Leopard.

mushkin **PerfectMatch**

Memory to match your system perfectly.

- ***Guaranteed, 100% Compatibility***
  Every PerfectMatch module is guaranteed to match or exceed the performance of the original memory installed at the factory.
- ***Save hundreds of dollars over the price of a new Mac***
  Mushkin PerfectMatch memory upgrades provide an easy and inexpensive way to optimize overall performance.
- ***Taking care of your company's road warriors?***
  Now you can cater to your laptop users' voracious appetite for RAM and enhance their mobile computing lifestyle.
- ***Your creative genius just met its match.***
  Upgrading with PerfectMatch memory increases performance in memory-hogging multimedia applications.

Now That Cat Can Run.

v i s i t u s o n l i n e a t w w w . m u s h k i n . c o m

North America

333 W Colfax Ave Ste 400
Denver, Colorado 80204
United States of America
tel: (800) 569-1868
fax: (303) 534-0827

Europe, Middle East & Africa

Raiffeisenstr.9
76872 Minfeld
Germany
tel: +49 7275-9888-13
fax: +49 7275-9888-69

Asia & Pacific

B-13-9, Megan Avenue II, No12
Jalan Yap Kwan Seng
50450 Kuala Lumpur
Malaysia
tel: +60-12-3157008

<m Get More.

courses to end users. To become an ACT, candidates must submit an application. Once the application is accepted, candidates must attend a Train-the-Trainer course and pass the associated trainer exam. For more details on becoming an ACT, visit: http://training.apple.com/act/

Apple Certified Master Pro and Master Trainer Certifications

Apple offers both end-user and trainer versions of the Final Cut Studio Master and Logic Studio Master certifications. Master certification recognizes your expertise with an entire product suite. Since workflow is such a significant component of both Final Cut Studio and Logic Studio, each curriculum is set up so that you must earn individual certification in multiple products in the suite to earn Master Certification, as outlined below. As exams are retired when a new version of the application is released, you will need to re-certify on the new versions to keep your Master Certification current.

To become a Final Cut Studio Master Pro you need to pass five exams:

Final Cut Pro 6 Level One

Final Cut Pro 6 Level Two or Color Level One

Motion 3 Level One or Final Cut Studio Motion Graphics Level One

DVD Studio Pro 4 Level One

Soundtrack Pro 2 Level One

To become a Logic Studio Master Pro you need to pass three exams:

Logic Pro 8 Level One

Logic Pro 8 Level Two

Soundtrack Pro 2 Level One

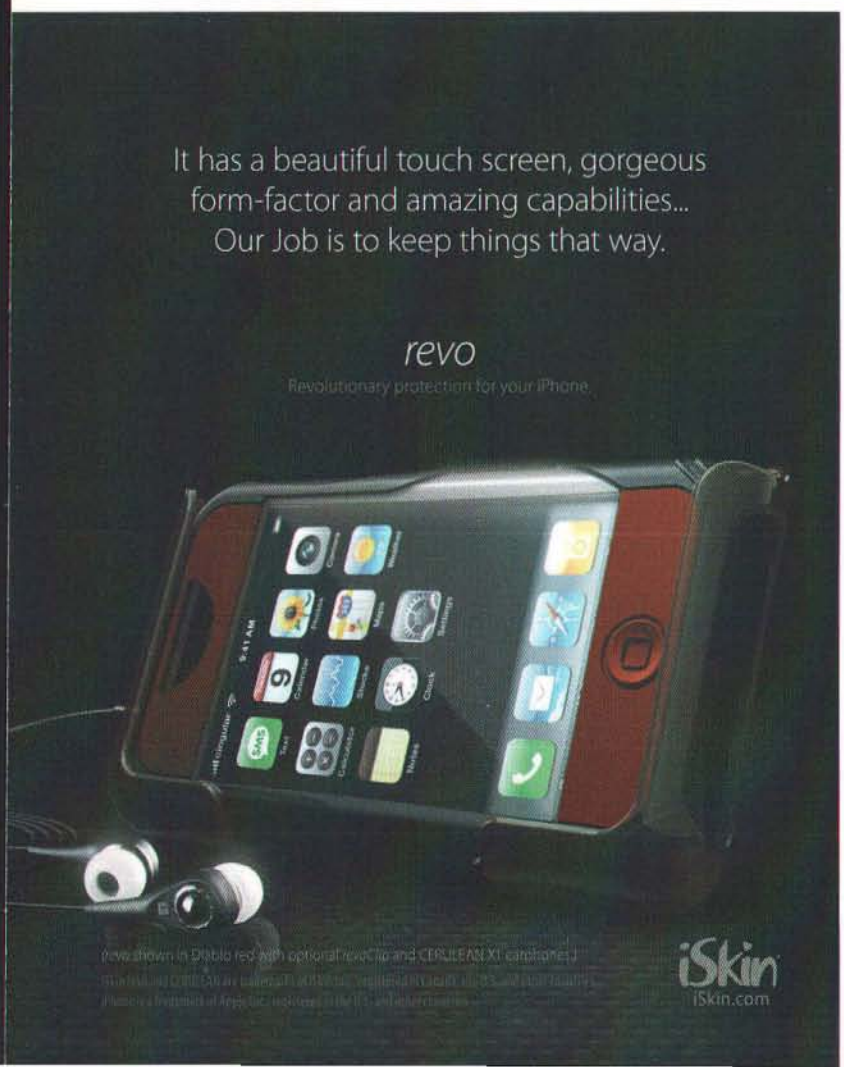
Remember all Level Two exams require attendance of the associated class to be eligible to take the test. Becoming a Master Trainer requires Trainer certification for each of the classes listed for Final Cut Studio or Logic Studio.

Preparing for the Exams

In previous articles, we reviewed some of the areas covered on the tests. Pro Apps exams focus more on application features than general technical knowledge. The best way to prepare for one of Pro Apps exams is to take the associated class at an AATC. You can find the nearest AATC at: http://training.apple.com/locations. You could also prepare by reviewing the Peachpit book for the particular course, but you will be more thoroughly prepared by participating in a class.

Next Article on the Road to Certification

In the fifth article in this series, we will provide detailed information about the requirements for the Apple Certified System Administrator (ACSA) 10.5 certification. We will discuss the topics covered on each of the exams required for ACSA certification, and what resources are available to help you prepare. Those resources of course will include Apple Authorized Training Center classes and books. We will also discuss how to become an Apple Certified Trainer for IT courses.



About The Author

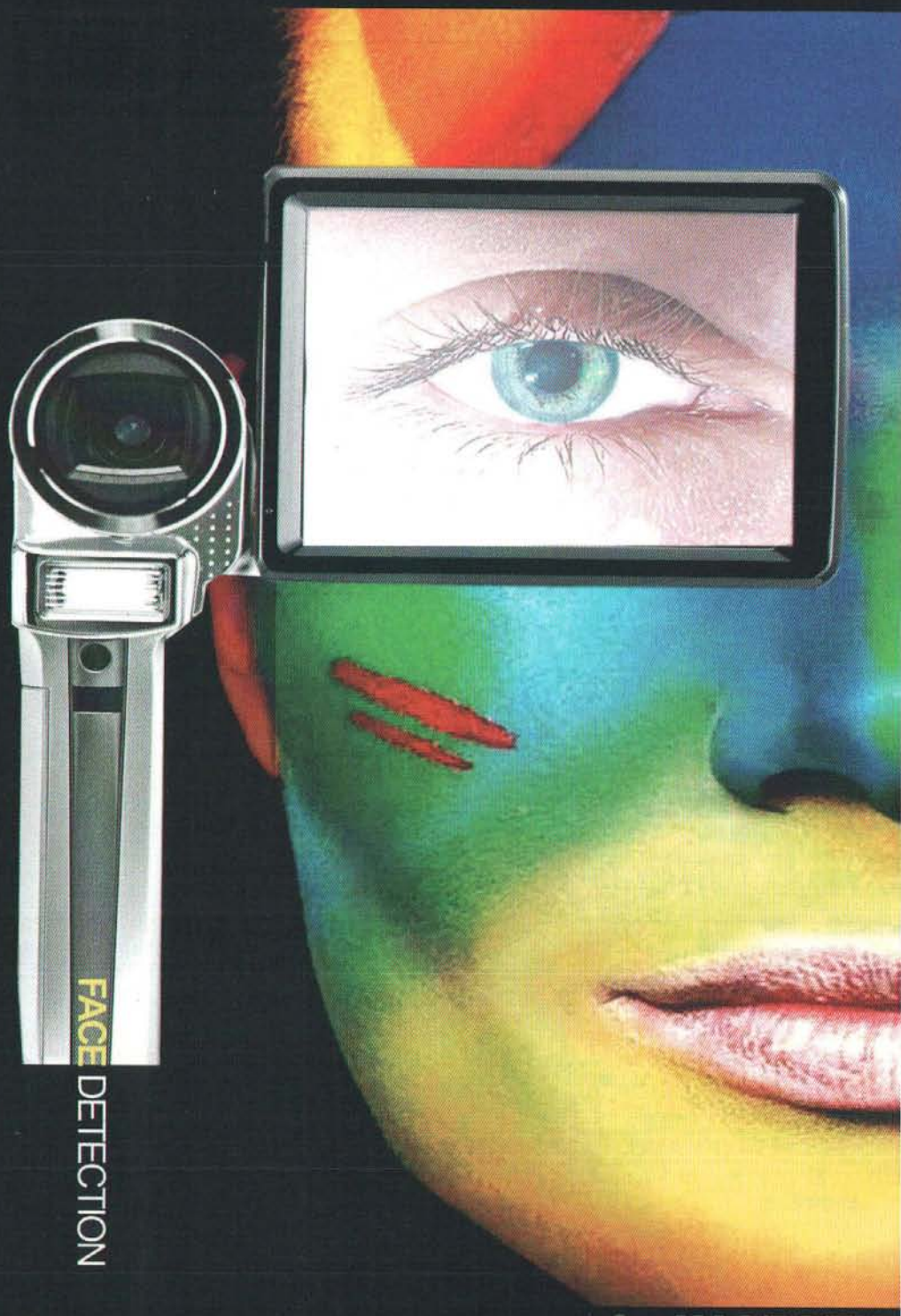
Doug Hanley owns MacTEK Consulting & Training, an Apple Authorized Training Center in Las Vegas, NV. His time is divided between teaching classes and wrangling servers. He has been providing support on the Mac since the early 90's. To track him down, go to http://www.mactektraining.com ***or email*** doug@mac-tek.com

MACTECH

Visit mactech.com for the latest Macintosh technical information.

supacam.com

The ONLY camera that streams DVD Quality Video across the internet



for MAC/PC/TV

supastream.com

"The full VGA streaming gives a picture **that brings Video sharing to a new height"**

Professor Norman Medoff author Portable Video ENG

Sharing with Git

Learn how to use Git to share your Xcode projects

by José R.C. Cruz

Introduction

CVS, Subversion, and Git let several users work the same project. Users can contribute their changes and avoid getting in each other's way. They can retrace their steps if they make a mistake. They can also review each other's changes with little effort.

But CVS and Subversion use a *central* repository, at risk to data loss without a good backup plan. Git, on the other hand, supports the use of *distributed* repositories. Each user keeps a separate repository for the same project. They then update each other's repositories at specific times in the project schedule.

This article shows how to use Git to setup a basic distributed repository. It shows how to move data between repositories. It also shows other ways to send data updates when those repositories are not available.

Planning Your Repository

Git lets you setup your project repository in many ways. The simplest way is to use a *central repository* (Figure 1). User A checks out a copy of the project on which he makes his changes. Then he commits his changes back into the repository, thus allowing user B to update her copy as well.

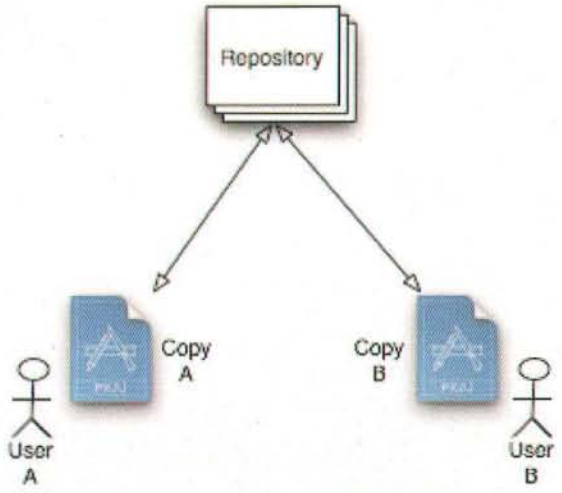


Figure 1. The centralized setup

But this setup, as stated earlier, has a set of problems. First, losing the repository means losing all project data. Second, users need constant access to the repository in order to commit their changes. Third, the repository can get large and unwieldy as time goes on.

A different way is to use a *two-tier setup* (Figure 2). Here, a *public* repository contains all the files needed by the project. Next, users A and B have their own *private* repositories, which are direct duplicates of the public one. User A makes his changes to the project and updates his repository. User B also does the same to hers. Then, at an agreed time, users A and B update the public repository with data from their private ones. They also work with the project head to resolve any conflicts that occur.

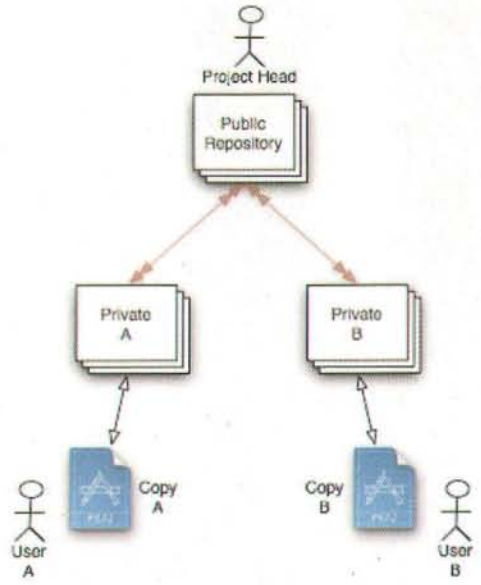


Figure 2. The two-tier setup

Since users A and B have their own repositories, they access the public one only when they need to. They are also free to try out new ideas, knowing that they will send only those changes that worked to the public repository. And if user

Photofont

More than just one face of the web

CRAFTED BY FontLAB

WWW.FONTLAB.COM

In the old days, if you were designing web pages, you had two choices: **a)** make the type look good using a bitmap — and lose the ability to have your text indexed, searched, copied, or spell-checked, or **b)** use a generic web font.

That's history. With **Photofont** technology, your website can boast impressive typography, yet keep all the advantages of standard HTML text. Test it yourself at **http://www.photofont.com**

WEB TYPOGRAPHY

Photofont® WebReady™ lets you format web headlines and other short web texts with any fonts available on your computer without damaging the original html. So you can still search and copy the text and virtually every user will see your website the way you intend it. And search engines such as Google will like your pages, too!

PRINT TYPOGRAPHY

Our Photofont® plugins for **Adobe Photoshop®, Adobe InDesign® and QuarkXPress®** allow you to use photofonts in any print publication as well as on web pages.

COLOR TYPOGRAPHY

But **Photofont®** technology allows you to use more than just the normal fonts. You can also use photofonts with full color or grayscale, gradients, transparency and cool effects.

With **BitFonter™**, you can make photofonts out of any digital image. Perfect for illustrators, scrapbookers, photographers or calligraphers.

FONT EDITOR HEADQUARTERS

Fontlab Ltd. is the world leader in font editors and converters: **TypeTool™, TransType™, Fontographer™, FontLab Studio™, AsiaFont Studio™, BitFonter™, ScanFont™, and more.**

VISIT US AT **WWW.FONTLAB.COM** AND **WWW.PHOTOFONT.COM**

Copyright © 2007 by Fontlab, Ltd. Inc. BitFonter™, FontLab®, Photofont™, TypeTool™, Fontographer™, FontLab Studio™, AsiaFont Studio™, TransType™ are trademarks of Fontlab Ltd. Inc. InDesign® and Photoshop® are registered trademarks of Adobe Systems, Inc. QuarkXPress is a trademark of Quark Inc.

A loses his repository, he can make a new one using the public repository.

A more radical way is to use a *round-robin setup* (Figure 3). First, user A creates a private repository for his project. Then he creates a public copy of that same repository. User B, who just joined in, gets a copy of user A's public repository. She then makes another copy to serve as her public repository. During the course of the project, user A sends updates from his private to his public repository. But he receives updates from user B's public one. The same steps also hold for user B.

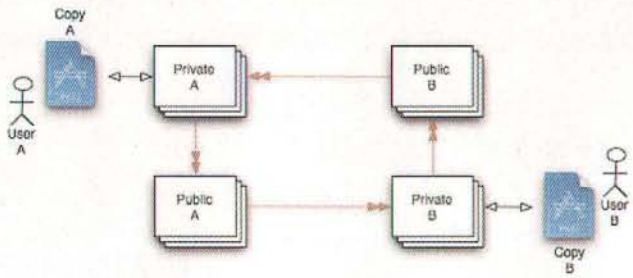


Figure 3. The round-robin setup

The round-robin setup removes the need for a central repository. It does, however, need more storage resources. It can also be more difficult to maintain when a lot of users are involved.

Enter The Daemon

You normally store the public repository on the network volume. How you access it depends on what server daemon you used. That same daemon also dictates the setting for the global option `–git-dir` or for the environment variable `GIT_DIR`.

For example, assume you place the repository for project `foobar` on the network volume `foobar.net`. If the server daemon is Apache, add the following URL in your Git commands.

```
git –git-dir=http://foobar.com/foobar/ …
```

If it is `rsync`, add the following lines to your `.bash_profile` file.

```
GIT_DIR=rsynch://foobar.com/foobar
Export $GIT_DIR
```

Git also supports the use of `https` and `ssh` as valid daemons. But the one we will look into is the git daemon.

The git daemon

The `git daemon` provides basic TCP/IP services for the Git repository (Figure 4). It listens to port 9418 and handles any SCM requests that appear on that port. It is easy to setup and use than most network servers. It does not yet support authentication, but this support may appear in future revisions.

Below is the basic syntax of the `git daemon` command.

```
git daemon options –base-path=volume_path white_list
```

The *`options`* argument consists of one or more flags to control

lucy didn't invent the Gateway.

she perfected it.

Introducing the future of payment gateway solutions.

Let Us Connect You: Cynergy Data's LUCY Gateway securely processes all forms of electronic payment—credit cards, PIN debit, gift cards, signature capture, checks and EBT.

Our reliable, flexible and secure gateway also gives you:

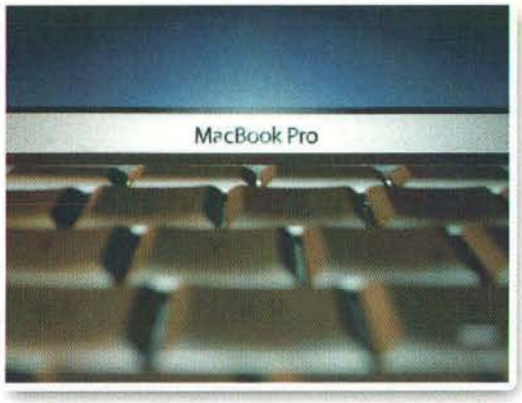
- Free software developer kit and integration support
- Transaction times of 3-5 seconds
- 24/7/365 technical support
- Intrusion detection and SSL encryption
- Residual Income for partners when your merchants use LUCY

cynergydata.com/gateway 866.414.3690

lucy Gateway let us connect you
A Cynergy Data Solution

Short Term Computer Needs?

- Latest Apple Technology
- Legacy Macs
- Windows Computers



- Rent One or Hundreds
- Audiovisual Gear
- Projectors, Plasmas, LCDs
- B&W & Color Copiers

Mac Rentals
Event Technology Services

- Name Brands
- Nationwide Delivery
- Professional Setup
- Onsite Technicians
- Same Day Service

- Compatibility & Compliance Testing
- Short Term Projects, Production Needs
- Trade Shows, Conventions, Corporate Events
- Trainings Sessions, Classrooms
- Fortune 100 and Educational P.O.'s Accepted!

Toll Free

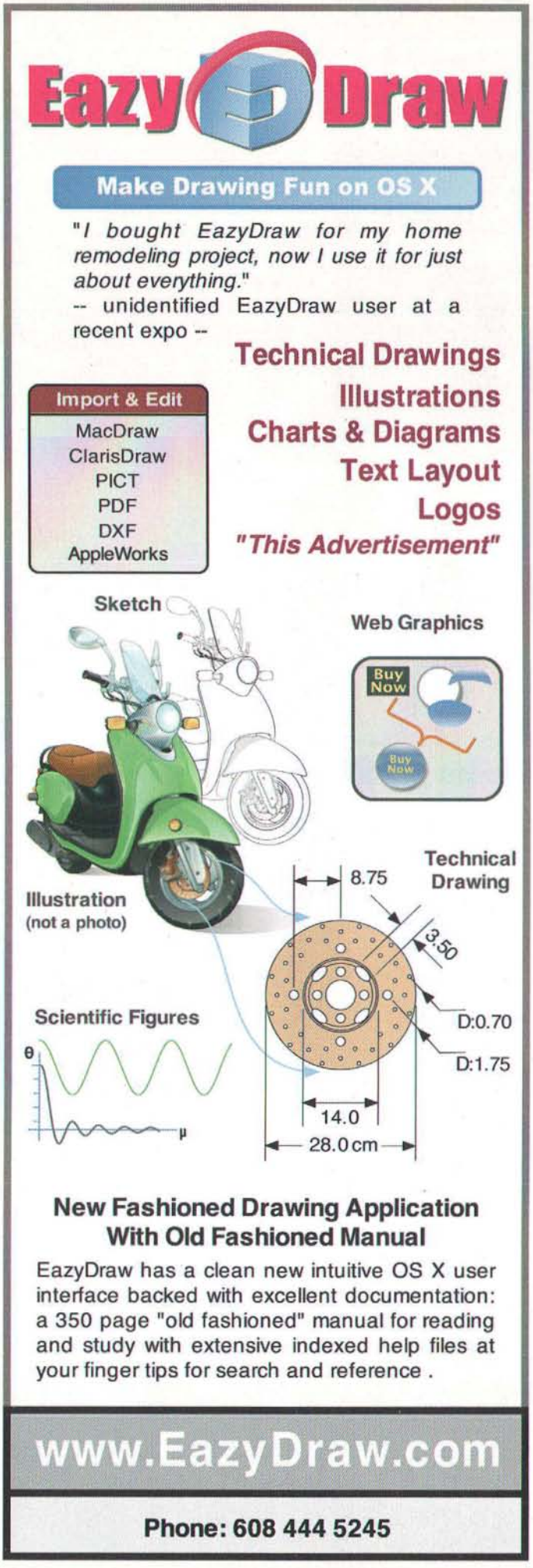
800-756-6227

Outside US/Canada: 858-454-8535

For Online Quote Request, Visit:

www.madcomputing.com

Largest Apple Rental Inventory in the U.S.!



the daemon. The `–base-path` option sets the path to the network volume holding the repositories. The *`white_list`* argument is a list of paths to each repository serviced by the daemon.

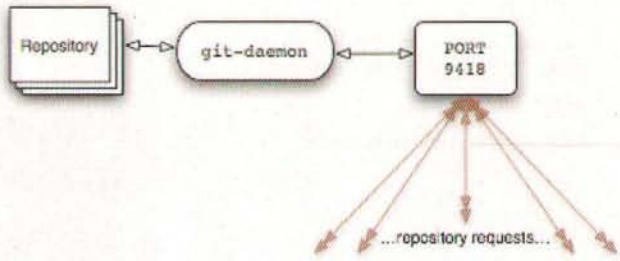


Figure 4. The git daemon

Launch the daemon

Assume that your project repository `foobar` is in the network volume `/Volumes/Server/Projects`. Also, assume that the volume uses the domain name `foobar.net`. To start the daemon, enter the following statement at the Terminal prompt.

```
git daemon –base-path=/Volumes/Server/Projects \
    /Volumes/Projects/foobar &
```

The `'&'` token tells the daemon to launch itself in the background. Otherwise, you will be unable to enter more command unless you stop the daemon with a `Ctrl-C` keystroke. To check if the daemon is running correctly, type the following statement at the prompt.

```
git ls-remote git://foobar.net/foobar
```

Git should then display a list of nodes and SHA1 keys as shown below.

```
b491b9426501f78575632588d2a86dbbb242df1d        HEAD
b491b9426501f78575632588d2a86dbbb242df1d
refs/heads/master
```

Now assume you want to include the repository barfoo to the daemon's white list. To do so, first stop the daemon using the `kill` command to. Then launch the daemon again as follows.

```
git daemon –base-path=/Volumes/Server/Projects \
    /Volumes/Projects/foobar /Volumes/Projects/barfoo &
```

When you enter the directory paths, make sure to type the paths *in full*. Avoid using shortcut tokens such as `'~'`. Also avoid adding a trailing `'/'` to each directory path. Either token will only prevent the daemon from finding the repository.

Starting the daemon from a Terminal session does have one problem. If you end the session, i.e. by closing the window, you *stop* the daemon as well. A better approach is to start the daemon using either `inetd` or `launchd`. For instructions on how to do so, see the references listed at the end of this article.

Configuring the daemon

You can change how the daemon behaves by using one or more options. For instance, the daemon sends its error messages to **stderr**, which is the Terminal window. To send those messages to the file **system.log** instead, add the **–syslog** option.

```
git daemon –syslog …
```

Also, the daemon listens to port 9418 for incoming requests. To use a different port, e.g. 9500, pass the new number using the **–port** option.

```
git daemon –port=9500 …
```

Make sure that the new port number is not used by other network services. To see if the number is available, check the **/etc/services** file as follows.

```
more /etc/services | grep 9500
```

By default, the daemon uses only those repositories in its white list that have the zero-byte file **gitdaemonexportok**.

To remove this limitation, add the **–export-all** option.

```
git daemon –export-all …
```

The daemon now handles all the repositories in its white list, *as long as* they have the subdirectories **.git/objects** and **.git/refs**. If these two subdirectories are missing, the daemon will return an error message. You can also add the zero-byte file to the repository by entering the following statement.

```
touch /Volumes/Projects/foobar/.git/config/git-daemon-export-ok
```

For a detailed list of other daemon options, read the following online document.

`http://www.kernel.org/pub/software/scm/git/docs/git-daemon.html`

Facing The Public

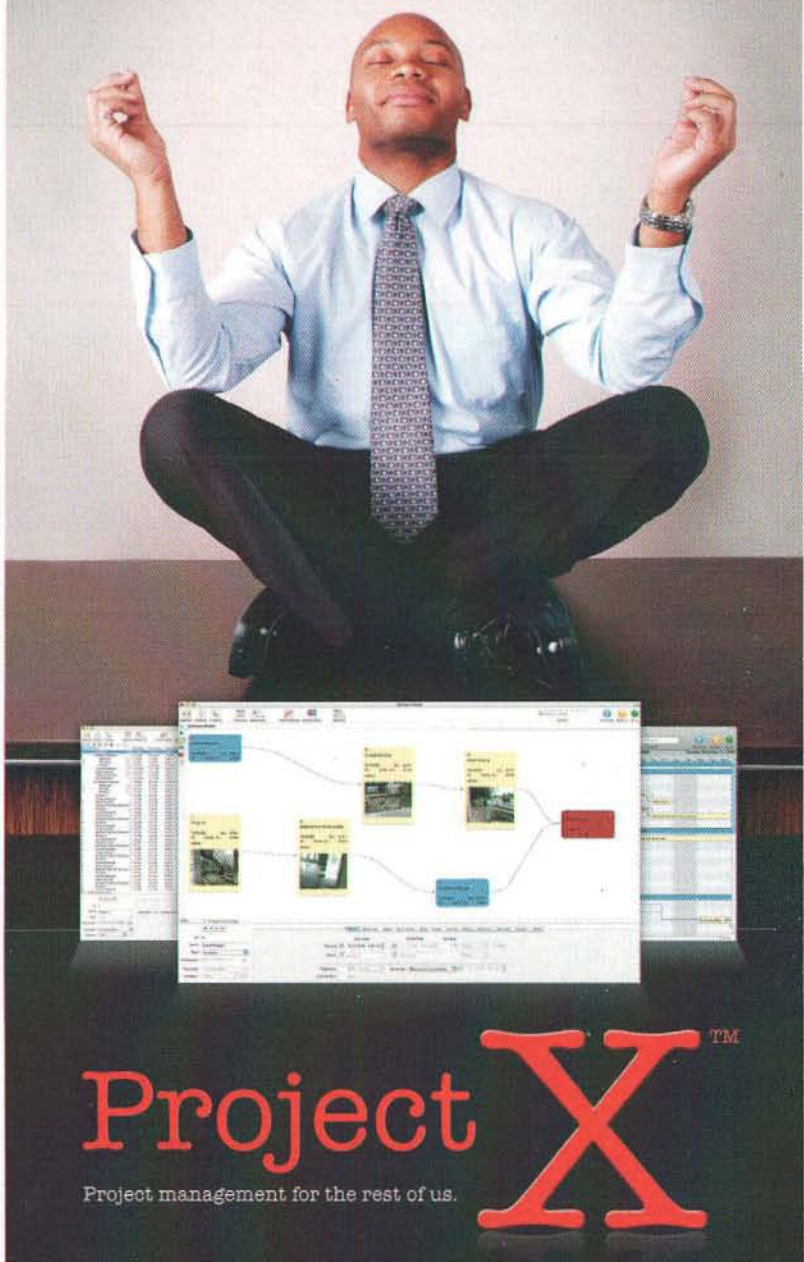
Before you let other users to access your repository, first make sure that its data is ready for public use. Check if the project compiles without any problems. See if the project has the minimum set of support documents such as a README and a version history file. Use the **git fsck** and **git gc** commands to remove unwanted data from the repository. Also, backup the repository before making it public.

These extra checks will help ensure a reliable and error-free repository.

Making clones

Use the **git clone** command to make a copy of the project repository. The command stores the copy in the location of your own choosing. It also updates the copy with data that links it back to the original.

Project management has never been so EASY.



Project X™

Project management for the rest of us.

While other project managers spend most of their time entering project tracking information, you use Project X's unique Web App to let your resources take care of the tedious data entry.

One click instantly incorporates timesheets, expense reports, and other task information, so you can spend time where it counts—managing your project!

Successful project management has never been so easy.

Macworld BEST OF SHOW 2SOXOF6

Version 1.3 Now Available
90 days of free telephone support

Experience Project X, download your free fully functional 30 day trial at www.ProjectX.com or call 954.927.6031

MARWARE

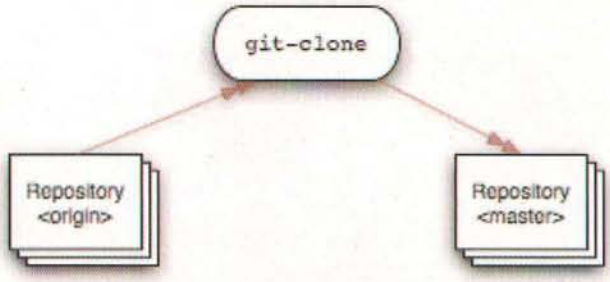


Figure 5. The git clone command

The basic syntax of the command is as follows.

```
git clone options repository_url destination_path
```

The ***repository_url*** argument is the URL to the repository being copied. The ***destination_path*** argument is the location of the copy. The ***options*** argument consists of one ore more settings to control the copy process.

Assume, for example, you want a copy of the project repository **foobar**. To make the copy, enter the **git clone** command as follows.

```
git clone git://foobar.net/foobar
/Volumes/Projects/private
```

First, Git sends a copy request to the daemon. The daemon responds by sending back the repository data. Next, Git stores the copy in the directory **/Volumes/Projects/private**. It then marks the copy as the *master*, the original as the *origin*.

Choosing which repository to copy depends on the project's state and setup. In an ongoing project, make a copy of public repository. The copy then serves as your private repository. In a new project, however, make a copy of your private repository. In this case, the copy itself becomes the public one.

Exchanging data

Use the **git pull** command to move data from the public repository to your private one. This command uses the public data to update your repository (Figure 6). It also updates your working copy, if present. It then reports any conflicts found during the update.

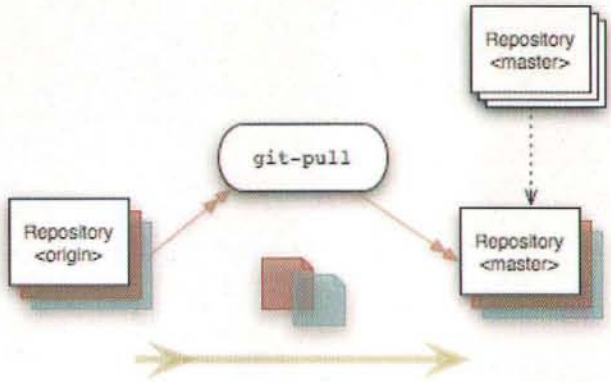


Figure 6. The git pull command

Faronics
DEEP FREEZE MAC™

Deep Freeze Mac delivers total system consistency to Mac computing environments. The need to perform rebuilds, re-image, or troubleshoot computers is eliminated with Deep Freeze Mac's reboot-to-restore technology. The result is a dramatic reduction in IT expenditures and a substantial increase in computer uptime. Deep Freeze Mac integrates into Apple Remote Desktop for enterprise deployment and management.

The **Bear Essential** for System Consistency

- Mac OS X Leopard Support
- Dynamic User Mapping
- Multiple Maintenance Schedules

Download a free, fully functional evaluation copy at **www.faronics.com**

Faronics
sales@faronics.com
1-800-943-6422

The only true graphical terminal server for Mac OS X

Aqua Connect Terminal Server is the only scalable enterprise grade solution that allows the Mac OS X platform to be deployed to multiple diverse devices, including Macs, PCs, and thin clients. Aqua Connect offers a feature set that truly simplifies OS X administration.

Mac

Universal

Aqua Connect

www.aquaconnect.net
(866) 543-AQUA (2782)

Copyright © 2008 Aqua Connect LLC. All rights reserved. Aqua Connect Terminal Server is a trademark of Aqua Connect LLC.In the US and/or other countries.Other names may be trademarks of their respective owners.

Below is the basic syntax of the `git pull` command.

```
git pull options repository_url references
```

The `options` argument consists of one or more settings to control the pull process. The `repository_url` argument is the location of the public repository. The `references` argument consists of the nodes or branches where the data is to be pulled. Leave this blank to retrieve only the latest updates.

Assume you want to update your private copy of the `foobar` repository. To start the update, enter the `git pull` command as follows.

```
git pull git://foobar.net/foobar
```

The above command gets the latest updates from the public repository. Then it merges them with the HEAD node on the private copy. To get updates from a specific node, e.g. `version_1`, enter the command as follows.

```
git pull git://foobar.net/foobar version_1
```

To display a diff report after data is retrieved, add a `–summary` option.

```
git pull –summary git://foobar.net/foobar
```

To keep the data from merging with your private repository, add a `–no-commit` option.

```
git pull –no-commit git://foobar.net/foobar
```

Use the `git push` command to move data from your private repository to the public one (Figure 7). If you are using the `git daemon`, make sure to tell the daemon to allow the push request. To do so, enter the following `git config` command on the *server side*.

```
git config daemon.receivepack true
```

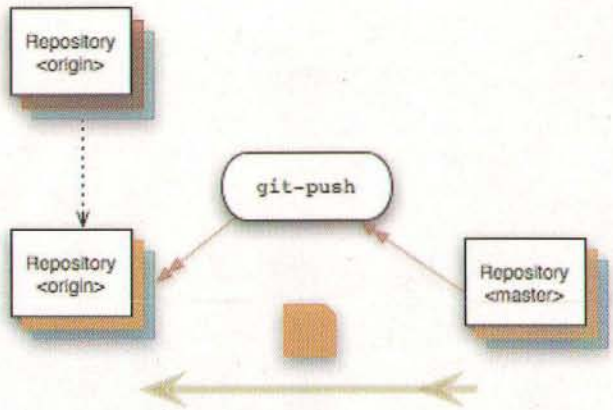


Figure 7. The git push command

The `git push` command has the same syntax as the pull command. But it has a much smaller set of options than the latter. Also, its use consists mostly of moving the latest updates to the public one. For instance, to update the public repository `foobar`, enter the command as follows.

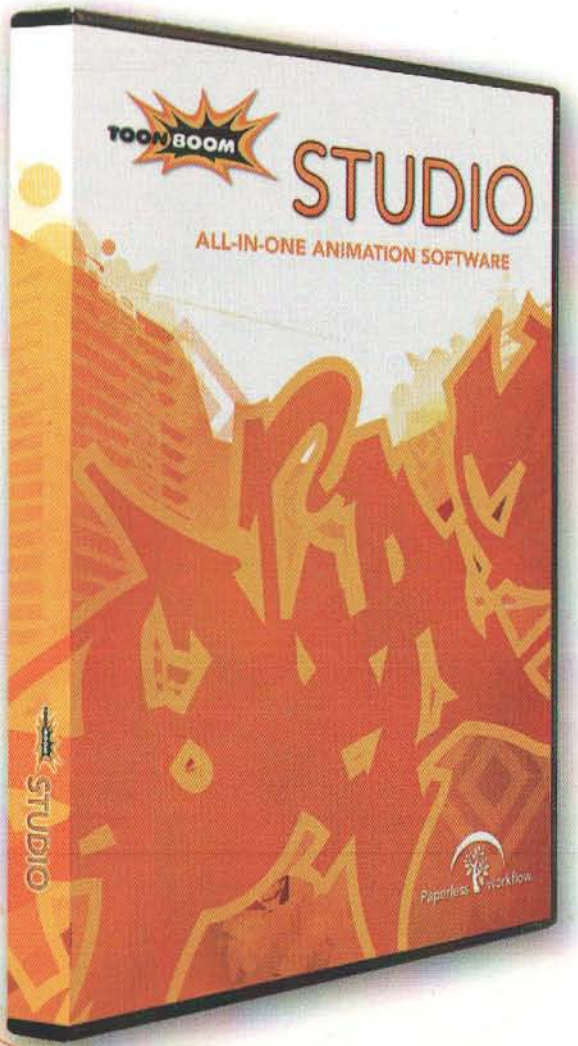
```
git push git://foobar.net/foobar master:master
```

Make sure to update your private repository first *before* updating the public one. You will find that it is much easier and safer to resolve any conflicts on the private copy than on the public one.

ANIMATE WITH EASE!

TOON BOOM STUDIO

ALL-IN-ONE ANIMATION SOFTWARE



The only start to finish animation software toolset designed for creative and professional users.

- Draw, scan or import artwork into system
- Paint artwork automatically
- Lip-sync mouth movements to voice tracks
- Animate in conjunction with sound, effects
- Access libraries to use and reuse content
- Create motion and camera moves in 3D
- Apply special effects
- View animation in real time
- Publish to TV, HDTV, video, web (SWF, QT)

Teacher's guide and animation lessons available

toonboom.com

The Daemon Is Down

There can be times when you will be unable to connect to the public repository. The server daemon may have crashed or terminated. The network may be unreachable. The repository itself may be corrupted. Or the server that hosts the repository may be getting upgrades.

For these cases, Git gives you two ways to still share your project changes with other users. You can use Git to make patches from your private repository. You can also use it to archive specific revisions in your repository. Then, you can send these patches or archives to those users via other means such as e-mail or portable media.

Making patches

Use the `git diff` command to make a patch for specific revisions of your project files (Figure 8). You can make a patch for the working copy of a project file against the one in the repository. You can make another patch for two revisions of the project in the repository.

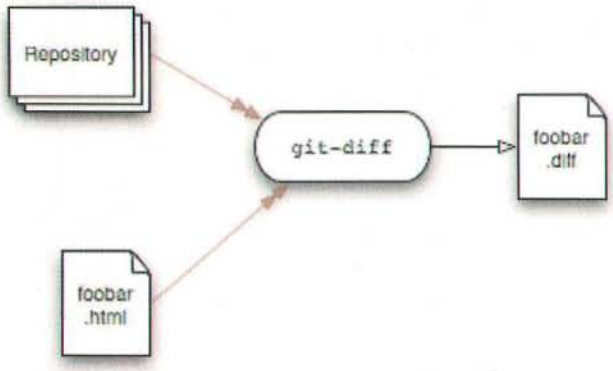


Figure 8. The git diff command

For example, to make a patch for the project file `foobar.html`, enter the command as follows.

```
git diff foobar.html > foobar.diff
```

Git first compares `foobar.html` with its latest revision in the repository. Then it stores any changes it finds in the patch file `foobar.diff`. To make a patch against the second latest revision of `foobar.html`, enter the command as follows.

```
git diff HEAD^ foobar.html > foobar.diff
```

To make a patch for the latest and second latest revision of project `foobar`, enter the command as follows.

```
git diff HEAD HEAD^ > foobar.diff
```

To make a patch for a binary file such as `foobar.jpg`, add a `–binary` option to the command.

```
git diff –binary foobar.jpg > foobar.diff
```

To make a patch for more than one file, enter their names as a space-delimited list.

```
git diff –binary foobar.html foobar.jpg > foobar.diff
```

Notice that the above example uses the `–binary` option since at least one of the files is a binary file. You can exclude this option if all the files are text files. Also, patch files use either a `.diff` or `.patch` suffix. But you can use your own suffix as long as you inform this to your fellow users.

Use the `git apply` command to update your repository with a patch. But check first to see if the patch is for a group of text files or for a group of text and binary files. If the patch is for *text files only*, enter the command as follows.

```
git apply foobar.diff
```

If it is for binary files only or for text and binary files, add a `–binary` option to the command.

```
git apply –binary foobar.diff
```

Either way, Git will use the patch to update the right file. It will, however, leave the updated files *uncommitted*. You must verify and commit the updated files manually. Doing so will protect your repository from unwanted changes.

Making archives

Use the `git archive` command to archive revisions of a project in the repository (Figure 9). This command can make two types of archives: *tarballs* and *zip files*. Choose the one most suited to your needs.

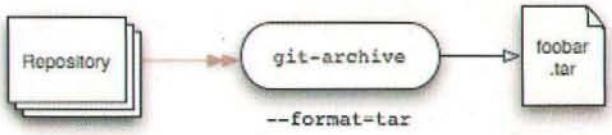


Figure 9. The git archive command

The basic syntax of the `git archive` command is as follows.

```
git archive options –format=type revision > archive_path
```

The `options` argument consist of one or more flags to control the archival process. The `–format` option selects the archive format. The `revision` argument is the project revision to be archived. And the `archive_path` argument is the name and path of the archive.

Assume you want to archive the latest revision of the project repository `foobar`. To make a tarball archive, enter the command as follows.

```
git archive –format=tar HEAD > foobar.tar
```

To make a zip archive, change the `–format` option as follows.

```
git archive –format=zip HEAD > foobar.zip
```

Notice that both examples has the archive's suffix set explicitly. Make sure to do so as the command itself will not set the right suffix for you.

To display a list of files added to the archive, add a `–verbose` option.

```
git archive –format=tar HEAD –verbose > foobar.tar
```

To add a prefix, e.g. `foo_`, to each archived file, use the `–prefix` option.

```
git archive –format=tar –prefix=foo_ HEAD > foobar.tar
```

Finally, if you get an archive from another user, use either the `tar` or `unzip` tool to open the archive. But make sure to

The killer app for the Mac is better than ever!

Parallels® Desktop 3.0 for Mac is here.

With Parallels® Desktop 3.0 for Mac, you can:

- Automatically open Windows files with Mac software and Mac files with Windows software
- 'Bulletproof' your virtual machine from Windows system crashes and malware
- Run selected Windows-only 3D games and applications with 3D graphics support
- Browse through Windows files and folders without launching Windows

SWSOFT PARALLELS®

Parallels® Desktop 3.0 for Mac provides even more groundbreaking features and capabilities than previous versions. Already the top-ranked software on PC World's "Best 100 Products of 2007" list, Parallels® Desktop continues to drive innovation in desktop virtualization.

Want better Windows and Mac integration? Want to run the hottest PC games and graphics software? Worried about security and system mishaps? Parallels® Desktop 3.0 for Mac includes 50+ new features and enhancements, including a number of integration features and virtual machine utilities unavailable anywhere else.

To discover why Parallels® Desktop is the leading solution for running Windows on a Mac, visit www.parallels.com or call us at 1 (425) 282-6405.

PC World 100 BEST Products of 2007 — Macworld BEST OF SHOW 2007

Download a trial today at www.parallels.com/download

Website – www.parallels.com Email – sales@parallels.com Phone – 1 (425) 282-6405

Merlin 2

Project Management with a bit of Magic!

Just three out of hundreds of features:

Network-based Project Management

Collaborate with others on the same project over the network. Just with a single mouse click.

Automatic sync to iCal

Sync your projects to iCal and then onto your iPhone or any other mobile device.

Professional Cost Calculation

Define Budgets top-down or bottom up and compare them to planned vs. actual costs.

ProjectWizards

Merlin 2 is built from project managers for project managers.

Version 2.5 out now!

Get your free demo version now!
www.merlin2.net

do so in a directory separate from your working copy. That way you will avoid overwriting your project files by accident.

Concluding Remarks

Git makes it easy for several users to share their project data with each other. It supports most server protocols, as well as provides its own server daemon. It can move data from one repository to another, or copy a specific repository. It can make patches or archives for specific project revisions.

These sharing features makes Git well suited for handling distributed setups. But Git can also handle a centralized setup, which may suit certain projects. This flexibility is one of the reasons why Git is getting a lot of notice from the developer world.

Bibliography and References

Marczak, Edward. "launchd: Judge, Jury, Executioner". *MacTech Magazine*, XXI:6. Xplain Corporation. Online: http://www.mactech.com/articles/mactech/Vol.21/21.06/launchd/index.html.

Fields, J. Bruce. "Sharing development with others". *Git Users Manual*. 2007 Aug 11. From the Linux Kernel Archives. Online: http://www.kernel.org/pub/software/scm/git/docs/user-manual.html#sharing-development

Hamano, Junio C. "Repository Administration". *Everyday Git with 20 Commands or So*. 2007 Nov 03. From the Linux Kernel Archives. Online: http://www.kernel.org/pub/software/scm/git/docs/everyday.html

About The Author

JC is a freelance engineering writer who lives happily in North Vancouver, British Columbia. He is a regular contributor to MacTech and REALbasic Developer. He also spends quality time with his nephew. He can be reached at anarakisware@gmail.com.

MACTECH® Magazine

Get MacTech delivered to your door at a price FAR BELOW the newstand price. And, it's RISK FREE!

store.mactech.com/riskfree

BookEndz®

Docking Stations for Apple Computers

DOCKING STATIONS FOR APPLE COMPUTERS

Convert your MacBook Pro®, MacBook®, or PowerBook®
into a desktop in seconds
without misplacing cables or damaging connectors.

- Aluminum Plate helps in cooling of notebook
- Connectors are routed to rear of Dock
- MagSafe® Housing prevents accidental disconnect

17" MacBook® Pro Docking Station

13.3" MacBook® Docking Station

- Also available in Black
- Built in DVI/VGA full size
- 5 USB 2.0 compliant ports
- Gigabit Ethernet RJ45

(4 port powered or unpowered hub, AC/DC adapter included)

15.4" MacBook® Pro Docking Station

15" PowerBook® G4 Docking Station

17" PowerBook® G4 Docking Station

12" PowerBook® Docking Station

Visit our website for latest product announcement www.BookEndzdocks.com

BookEndz®
Manufactured by Olympic Controls
1250 Crispin Drive • Elgin, Illinois 60123 • USA
Phone: 847-742-3566 • Fax: 847-742-5686 • Toll Free: 888-622-1199
E-mail: Sales@BookEndzdocks.com

"Image"ine That!

A fresh look at creating deployment images

By Philip Rinehart, Yale University

MacEnterprise.org
Mac OS X enterprise deployment project

IMAGES?

In the System Imaging and Deployment Power Tools session at Macworld this year, there was a large amount of time spent discussing system imaging. How does it work? What are best practices? Are there best practices? How do you deal with new hardware, and more were asked. I thought it would be useful this month to reexamine the practice of image creation. Let's start by looking at what we mean by an image.

THE "IMAGE"

In traditional Macintosh system administration, an image is a base operating system plus some amount of software that is applied to one or more machines. In general, the nirvana has been to create a single image that can be applied to many different models of Mac machines. Is this goal achievable, or is it like Buddhism, in which the journey is more important?

Rule 1: currency

The first rule of system imaging is that an image **must** be created from the most current model of Macintosh in any deployment. Typically this image is created from the installation media that was shipped with the hardware. For example, if your deployment includes a quad core Intel machine, then create the image from that disc. Is this rule inviolable? No.

How do you update an existing image? Wait until the dot release update. That is, if your image was created on 10.4.10, update your image for new hardware at 10.4.11. There is some debate about using a combo updater or a delta. Generally, I prefer to use the combo updater, as each and every update from the Gold Master is applied. This method can be used to update existing images pretty reliably. There are of course exceptions, but now that a universal operating system is available, Leopard, this method should work in most cases.

Rule 2: User templates

The second rule of thumb in system imaging is how to create a default template. Note that this method applies to local accounts, not network accounts. First, create a template user using system preferences. Next, login as that user, and run each application that it is important to configure. Common configuration items include web browsers, word processing applications, or site-specific applications that need consistent settings. There are a couple of gotchas though.

First, for the template user, don't store anything in the keychain. When the template is copied over any item stored in the keychain will be inaccessible to the new account. It is safe to delete the keychain after finishing custom configuration as well. Secondly, be certain to set the proper downloads folder for each web browser. In Tiger, if the Safari download folder were not set, a copied template would contain an inaccessible path. The last gotcha is for any preferences that are machine specific. This type of preference is usually stored in ~/Library/Preferences/ByHost. Common items here include iTunes preferences, screensaver preferences, and others. A hardware address is embedded in the preference file name. It can be corrected with a loginhook.

Rule 3: Cleanup

Before applying any image, it is important to do some basic cleanup. What should be cleaned up though? There are a few obvious things to remove for initial cleanup. Get rid of any Cache files stored in /Library/Caches, and /System/Library/Caches. Next, remove both swap files and sleep images. These are located in the directory /var/vm, and can be significant in size. One other cache can also be removed, the BootCache.playlist located in /private/var/db.

What about other types of files? Generally, I would recommend moving the Network Interfaces plist in /Library/Preferences/SystemConfiguration. In general, these files are machine specific, and will get recreated by the operating system on first boot. This also removes any possible conflict if an image has a different network interface configuration then its target.

System wide, these files are ignored by Time Machine, and can probably be eliminated from any image.

```
/.Spotlight-V100
/.Trashes
/.fseventsd
/.hotfiles.btree
/Backups.backupdb
/Desktop DB
/Desktop DF
/Network/Servers
/Previous Systems
/Users/Shared/SC Info
/Users/Guest
/dev
/home
/net
/private/var/db/Spotlight
/private/var/db/Spotlight-V100
```

Note that most of these files are fairly logical to be excluded. The operating system will recreate any of these files at first boot.

Clean up of the User Template can also be a little more complete as well. Here are the files that can be excluded from the User Template. Most are related to browsers and rss feeds.

```
Library/Application Support/MobileSync
Library/Application Support/SyncServices
Library/Caches
Library/Logs
Library/Mail/Envelope Index
Library/Mail/AvailableFeeds
Library/Mirrors
Library/PubSub/Database
Library/PubSub/Downloads
Library/PubSub/Feeds
Library/Safari/Icons.db
Library/Safari/HistoryIndex.sk
```

The last thing to eliminate is any Log files, both in /Library/Logs and /private/var/log. No need to have any of these items on an image! Note that the cleanup process is best scripted, as no one really wants to remember all of these steps!

APPLE SOFTWARE RESTORE

Now that you have created your "perfect" image, it is time to get it ready for deployment. Are there best practices here? I think so. First, it is best to boot from an alternate volume. This volume can be a separate partition, or an external drive. Once booted the steps are pretty straightforward.

1. Open Disk Utility
2. Select the drive or partition that is the model for imaging
3. Select "Image from Folder" and select the hard drive and wait. A long time.
4. Select "Scan Image for Restore" and scan the newly created dmg file.

That's it! You then have an image that is ready for deployment via whatever method you have in place to put an image on a machine via multicast ASR, NetRestore, or any other method you have of getting the image on the machine. One important note though, it is generally best to have at least **twice** the amount of space needed to create an image on your external drive. If your drive has any less than that, the imaging operation may fail.

THE FUTURE

The method I have just described is the "Classic" way of creating methods and is fairly tried and true. However, it isn't really very scalable, or flexible as the image is a point in time snapshot. The way of the future can be seen in two areas, the new System Imaging Utility in Leopard and InstaDMG from afp548.com. Both take the idea of monolithic image creation, and move it to a more modular approach. Ultimately this approach is far more flexible, allowing updating any image at a moments notice. It also makes it easier to be extremely flexible and adaptable, which is a good thing in today's fast moving environments. It was good to see all at Macworld this year, until next month see you on the lists!

About The Author

Philip Rinehart is co-chair of the steering committee leading the Mac OS X Enterprise Project (macenterprise.org) and is the Lead Mac Analyst at Yale University. He has been using Macintosh Computers since the days of the Macintosh SE, and Mac OS X since its Developer Preview Release. Before coming to Yale, he worked as a Unix system administrator for a dot-com company. He can be reached at: philip.rinehart@yale.edu.

The MacEnterprise project is a community of IT professionals sharing information and solutions to support Macs in an enterprise. We collaborate on the deployment, management, and integration of Mac OS X client and server computers into multi-platform computing environments.

by Dave Dribin

Building on a Solid Foundation

Exploring the Foundation framework

The Foundation Framework

Now that we've gone over the basics of the Objective-C language, including classes, inheritance, and memory management, we can start to write some real code. One of the great things about developing for Mac OS X is that in addition to getting a nice language to use, you also get a lot of reusable code to use in your own applications. This can save you lots of time because you can reuse what's already present in the system.

Reusable code on Mac OS X is often grouped together in a package called a *framework*. A framework is similar to a dynamically linked library (DLL) on Windows or a shared object (.so) on most other Unix platforms in that it contains shared code that may be used by multiple applications. Frameworks have one important difference: they may contain more than just code. Typically they will contain the header files needed to use the shared code, but they can also include other resources, such as images or sounds. Mac OS X ships with many frameworks for a number of purposes, including text manipulation, graphics, sound, and networking. All the system-supplied frameworks are located in the `/System/Library/Frameworks` directory. Some of these are C-based while some are Objective-C. The main Objective-C framework that provides classes used in *all* Objective-C applications is called the *Foundation framework*, or just *Foundation* for short. You've already been using part of the code in Foundation: `NSObject`. In this article, we'll be exploring some of the other popular Foundation classes.

Strings

A string is a collection of characters used to represent human text. We've been using strings when we use the `printf` function:

```
printf("Hello world\n");
```

The text between, and including, the double quotes is called a string, i.e. `"Hello world\n"`. We haven't really gone over how strings are implemented in C, so let's dive down a bit deeper.

Strings in C are an array of characters. The built-in type for text characters in C is `char`. The `char` type holds a signed integer between -128 and 127. You can assign a single character to a variable of type `char` using single quotes, and you can print out an individual character in `printf` with the `%c` formatting specification:

```
char letterA = 'A';
printf("Character: %c\n", letterA);
```

The output for this would be:

```
Character: A
```

So, if a string is an array of characters, you may think you would declare the array for "hello" as follows:

```
char hello[5];
hello[0] = 'h';
hello[1] = 'e';
hello[2] = 'l';
hello[3] = 'l';
hello[4] = 'o';
```

This is close, but there's one major issue. There's no way to determine the length of an array in C, thus there's no way to determine the end of a string. Those clever C designers thought up a way around this by using what's called the *null character*. The null character has the integer value of zero and may be entered directly with single quotes using backslash zero, `\0`. In C, strings *must* end with a null character, thus the correct way to define an array for "hello" is:

```
char hello[6];
hello[0] = 'h';
hello[1] = 'e';
hello[2] = 'l';
hello[3] = 'l';
hello[4] = 'o';
hello[5] = '\0';
```

Because the last character in the array is the null character, strings in C are called *null-terminated strings*. The standard C library has many string functions. One such function is `strlen`, which returns the length of the string. There are many more, but we won't be covering them here because, as we will see, Objective-C handles strings differently than C.

C provides a shorthand notation for initializing arrays in one step, so we can alternately initialize the array:

```
char hello[] = {'h', 'e', 'l', 'l', 'o', '\0'};
```

This syntax allows us to drop the array size, as the compiler can figure it out for you. Even though this syntax is better than before, it's still cumbersome. That's where the

Revolutionize your Lifecycle Management Strategy with LANrev!

True Cross-Platform Lifecycle Management has arrived!
Do you need to manage PCs with a Mac? Macs with a PC?
LANrev is the first client management solution
with true cross platform functionality
for today's heterogeneous client environments.

Disk Imaging
Asset Inventory
Software Distribution
License Management
Patch Management
Remote Management
Role-Based Administration
Change and Configuration Management
LANrev TheftTrack Client Recovery

LANrev is a scalable multi-platform solution, which can operate with equal efficiency on a single server deployment in the small to medium enterprise, or in a distributed architecture, managing thousands of desktops in a large enterprise.

LANrev's modern, modular platform allows seamless integration into other desktop management tools such as Microsoft's SMS, enabling your organization to leverage previous investments while maximizing current and future efficiencies through the use of innovative new technology such as LANrev. The best part? You can take advantage of your current hardware infrastructure. Unlike the other guys, LANrev does not have massive, dedicated server requirements and you won't need a team of overpaid consultants to deploy and use LANrev.

See for yourself how you can increase your IT efficiency and performance while lowering your cost by using LANrev in your computing environment. Schedule a free web demo and trial license by calling (214) 459-0136 or visiting www.lanrev.com.

Full Mac OS X 10.5 Leopard Support!

LANrev

Mac Universal for windows

double quotes come in. We can use double quotes to represent the exact same thing:

```
char hello[] = "hello";
```

The double quote syntax is called a *C string literal*. Just keep in mind that under the hood, string literals are still just null-terminated `char` arrays. Remember, too, that arrays are nearly the same as pointers in C, so yet *another* way to write this would be:

```
char * hello = "hello";
```

This `char *` type is the way you will see most strings declared in C. Often you will see `const char *`, too:

```
const char * hello = "hello";
```

The `const` keyword means that the contents of the string are constant and may not be modified. To print null-terminated strings using `printf`, use the `%s` formatting specification:

```
printf("Say: %s\n", hello);
```

While the null character and double quote syntax solve many issues of strings in C, they have one serious weakness that's not so easy to overcome: Unicode.

Unicode

Above, I mentioned that the `char` type holds an integer between -128 and 127. This means that every character needs an equivalent number value. The translation between number and text character is called an *encoding*. Back when C was invented, the most popular encoding was called American Standard Code for Information Interchange, or *ASCII* for short. ASCII encoded the English alphabet in upper and lower case, the digits 0 through 9 and a few other characters used for controlling teletype terminals. ASCII only has 127 characters specified, which is perfect for the `char` type. The problem with ASCII is that it only works for the English alphabet. It doesn't provide a way to represent accented characters or non-English alphabets, such as Russian, Greek, or any of the Asian languages, including Chinese or Japanese.

To solve this problem, computer scientists from around the world came together and created a master list of all human characters on Earth. The result is called Unicode, specifically the Universal Character Set. As you may imagine, the number of characters far exceeds the 255 available numbers of the `char` type. But these computer scientists were really smart. They created multiple Unicode encodings that map the Unicode code points (the Unicode terminology for character) into different tables. One such encoding is called UTF-8, and it's specifically designed for ASCII-based null-terminated systems, like C strings.

While using UTF-8 in C strings makes it *possible* to use Unicode in C, it's far from ideal. Many standard C functions don't work quite right with UTF-8, and dealing with UTF-8 for lots of string data can be slow. Because of this, Objective-C strings are not based on C strings.

By the way, fully covering Unicode and the different Unicode encodings would be an article in its own right. I'm only covering the basics needed to understand strings in C and

BMS

THE LAW OFFICE OF
BRADLEY M. SNIDERMAN

Helping clients with their software legal issues.

- Trademark and Copyright Registration
- Trade Secret Protection
- Licensing and Non Disclosure Agreements
- Assist with Software Audits

I am an attorney practicing in Intellectual Property, Business Entity Formations, Corporate, Commercial and E-commerce Law.

Please give me a call or an e-mail. Reasonable fees.

23679 Calabasas Rd. #558 • Calabasas, CA 91302
PHONE 818-706-0631 FAX 818-706-0651 EMAIL brad@sniderman.com

Objective-C. If you want to learn more about Unicode, there are plenty of good resources on the Internet.

Objective-C Strings

To get around many of the limitations of C strings, Objective-C includes its own string class called `NSString` as part of the Foundation framework. You can create a new `NSString` instance from a UTF-8 encoded C string using the `stringWithUTF8String:` method:

```
NSString * hello =
    [NSString stringWithUTF8String: "hello"];
```

Remember from our previous article on memory management that this class method creates an autoreleased instance of `NSString`. This means you don't have to worry about retaining or releasing it, so long as you are finished using it before the autorelease pool is released. There's a corresponding instance method constructor you can use, if you don't want an autoreleased object:

```
NSString * hello =
    [[NSString alloc] initWithUTF8String: "hello"];
// Must call [hello release] when finished
```

While these are very handy for using the C string literals we've already been using, it's a bit long-winded for regular use. Thankfully, Objective-C also has its own syntax for string literals. It uses double quotes like C strings, but it uses an at sign ('@') before the first double quote:

```
NSString * hello = @"hello";
```

Notice that using the Objective-C syntax results in an instance of the `NSString` class. String literal instances are not autoreleased, but you shouldn't release them, either. They are allocated automatically before the `main` function is called and are automatically released when your application exits.

The nice thing about Objective-C strings is that they are full-blown objects. This means you can call methods on string instances. For example, to get the length of the string, you would use the `length` method:

```
printf("NSString length: %d\n", [hello length]);
```

If you want to get a UTF-8 string for use with C functions, you can use the `UTF8String` method:

```
printf("Say: %s\n", [hello UTF8String]);
```

Keep in mind the memory returned from `UTF8String` is also autoreleased. If you need it to stick around longer than the current autorelease pool cycle, you'll need to copy it into a new C string.

This is just the tip of the iceberg on what you can do with `NSString`. It's a very powerful class and works well with Unicode. As such, there are many more methods available. Since strings are so heavily used, we will no doubt be using more of these methods. I will explain them as we encounter them, but consult the documentation for a list of all available methods.

NSLog

As you saw above, to print out an `NSString` with `printf`, we had to convert it into a UTF-8 string. Not only is this a bit of a pain, but it can also be inefficient to convert the string into a new encoding. Unfortunately, since `printf` was designed only for C code, it cannot natively handle `NSStrings`. The Foundation framework comes with its own printing function called `NSLog`. It works very similarly to `printf` in that you can give it a string to print; however, you must pass it an `NSString` instead of a `const char *` C string:

```
NSLog(@"Hello world");
```

The output is a little different than `printf`. First, it automatically includes a newline on the end, so you do not need to use `\n` as the last character. It also includes extra information, such as the date and time, before the message. Here's the output when I ran the code above:

```
2008-01-07 15:30:35.539 objcstrings[13448:10b] Hello world
```

You can also use all the `printf`-style percent formatting specifications, like `%d` and `%s`. But it also comes with a new formatting specifier for Objective-C strings: `%@`. You would use it like this:

```
NSString * hello = @"hello";
NSLog(@"Say: %@", hello);
```

The resulting output should be similar to:

```
2008-01-07 15:39:46.147 objcstrings[13486:10b] Say: hello
```

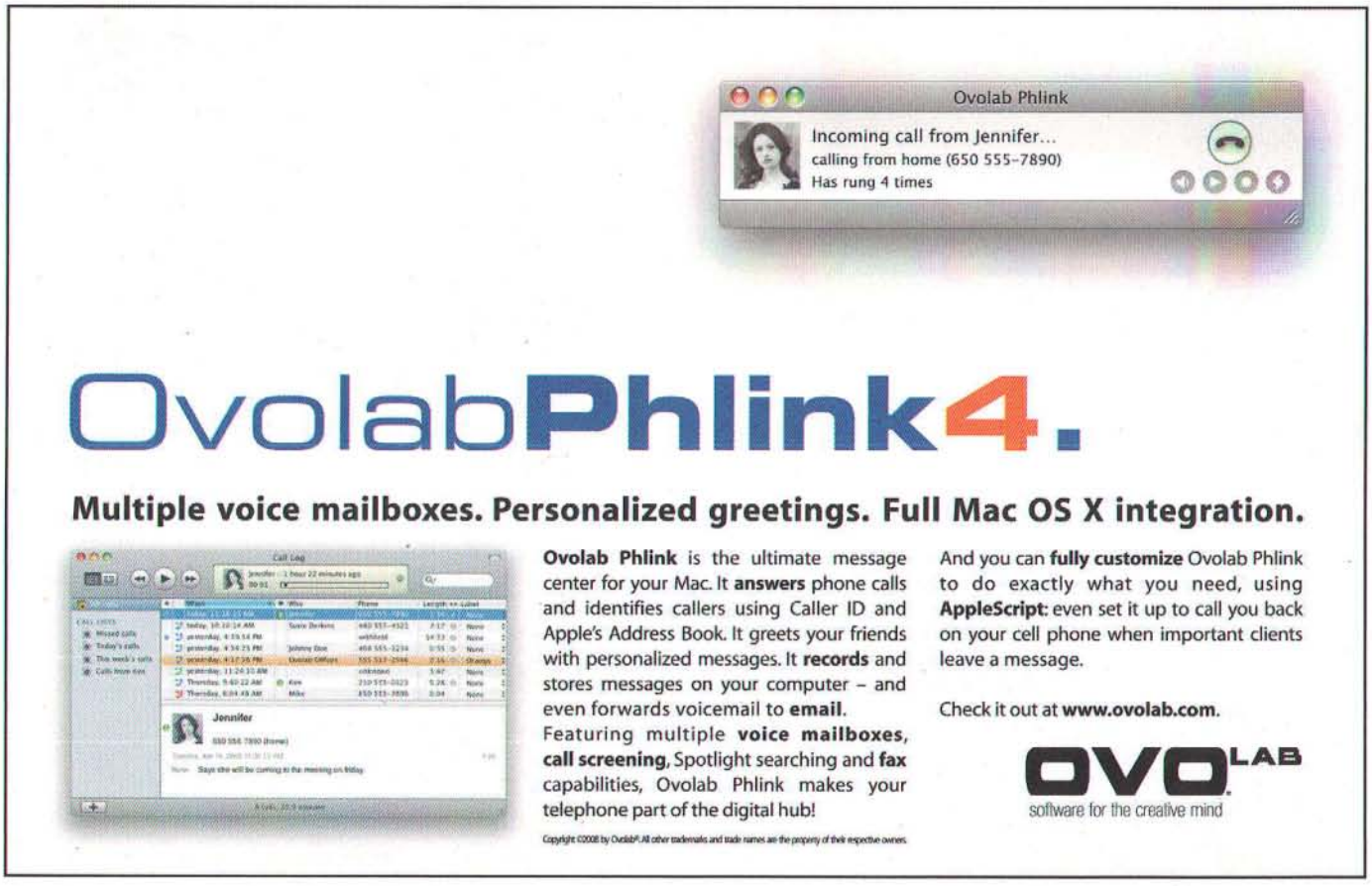
Even though `NSLog` prints extra stuff, you will often see it used in Objective-C code rather than `printf`. This is mainly because the extra stuff printed is helpful for debugging GUI applications. If you want more control over the output of your text, you'll have to use `printf` with the `UTF8String` method. Since we are still writing command line applications, for now, I will use `printf` as the output has a lot less clutter.

Modifying Objective-C Strings

An `NSString` instance is not modifiable, meaning you cannot change the text of the string. You can create a new string, but you cannot modify its contents directly. A fancier way to say "not modifiable" is *immutable*. Thus `NSString` instances are said to be immutable. There is a class called `NSMutableString` that creates a string whose contents may be modified directly. There is no shortcut way to create them, so you must use one of the constructor methods. One of the methods to change the string is the `appendString:` method, which adds text to the existing text:

```
NSMutableString * helloWorld =
    [NSMutableString stringWithString: @"hello"];
[helloWorld appendString: @" world"];
```

This results in the string `@"hello world"`. `NSMutableString` is a subclass of `NSString`. This means



you can call any method of an **NSString** such as **UTF8String**:

```
printf("Say: %s\n", [helloWorld UTF8String]);
```

Remember from our article on inheritance that you can use a subclass instance anywhere a superclass is used. Thus, you can pass in an **NSMutableString** anywhere an **NSString** is required.

Collections

While the string classes of the Foundation framework are extremely popular, the next most popular classes are called collections. Collections are classes whose sole purpose is to hold other objects. There are different collection classes depending on your needs, and we'll be looking at arrays and dictionaries.

Arrays

In previous articles, we've covered arrays in C. Arrays hold multiple values of the same type. C arrays are very limited in their functionality, though, and it's easy to use them incorrectly. The Foundation framework has a class for arrays named **NSArray**. It holds zero or more Objective-C objects and remembers their order. It is similar to an array in C, but it is a lot more flexible. To create an array you can use the **arrayWithObjects:** constructor:

```
NSArray * colors = [NSArray arrayWithObjects:
                    @"red", @"green", @"blue", nil];
```

This method creates an autoreleased array with three string elements. Note that the list of elements in the constructor is terminated with **nil**. It is important to not forget the terminating **nil**. If you do, you will most likely crash your program.

You can access individual elements of the array with the **objectAtIndex:** method. Just like C arrays, the index of the first element is 0, thus to access the second element, you'd use an index of 1:

```
NSString * green = [colors objectAtIndex: 1];
```

You can find out how many elements are in the array with the **count** method. Combining these two methods with a **for** loop, we print all the elements:

```
int i;
for (i = 0; i < [colors count]; i++)
{
    NSString * color = [colors objectAtIndex: i];
    printf("Color %d: %s\n", i, [color UTF8String]);
}
```

This should give you the following output:

```
Color 0: red
Color 1: green
Color 2: blue
```

NSArray objects are immutable. Just like **NSString**, there is a mutable subclass: **NSMutableArray**. A common

MacResource Computers and Service

Your iMac/eMac/Laptop Xserve & PowerMac Resource

Wireless Products

Airport Cards
Extreme: $69.99
Standard: $99.99
802.11N Upgrades
$79.99
Routers: $59.99
Express $99.99
Extreme $179.99
BlueToothUpgrade
$39.99/59.99/79.99

Parts, Parts, Parts

We have an extensive supply of cosmetic parts for iMacs. G4/G5 Towers, eMacs etc. LCD Panels for G4 iMacs G5 iMacs, PowerBooks and Displays.

Parts for Every Mac System!

Logic Boards

G3/G4 PCI/AGP : $129
G4 Gigabit/D Audio: $199
G4 Quicksilver: $349
G4 MDD: $489
G4 eMac Logics From $189
G4 iMac Logics From $199
G5 Towers $399/$499/$599
G4 Xserves $149-$299
G5 xserves $599

Power Supplies

G4 iMac 15/17/20: $79/$99/$119
G5 iMac 17/20: $149/$179
G5 Tower: $169/$199
G4 Qsilver/D audio:$179
G4 Gig E-net/MDD: $199
G4 AGP 208/237w: $179/$219
Logic board and power supplies require exchange.

Processors

Processors For G4s, G5 & Xserves
G4 466/733/800 $49/$149/$199
G5 1.6/1.8GHZ $399/$499
DUAL PROCESSORS ($ PER PROC.)
1.8/2.0/2.3GHZ 399/549/599
2.5DP/QP $699/$799
XSERVE PROCESSORS
G4 1.33GHZ DP $189
G5 2.0/2.3GHZ $229/699

Thousands of parts for all Mac systems!

1-888-Mac-Resource

www.mac-resource.com

Mac Systems

G4 466Mhz $199
G4 733/800Mhz $449/$549
G5 1.6/1.8Ghz $799/$899
G5 1.8/2.0Ghz DP$1199/$1299
G5 2.5/2.7Ghz DP $1499/$1599
G5 2.5Ghz QUAD DUAL DVI $1799
NEED G5 IMACS?
G5 1.6/1.8/1.9Ghz 17" $699/$799/$899
G5 1.8/2.0/2.1GHZ 20" $899/$999/$1099
EMACS GALORE, GREAT WORK STATIONS
700MHZ 256/40GIG/COMBO/17" $319
1.0GHZ 256/40GIG/COMBO/17" $349
1.25GHZ 256/40GIG/COMBO/17"$399
1.42GHZ 256/80GIG/SD/17" $499

New Systems Arriving Daily! Call for latest Stock.

WE HAVE G5 XSERVES AND RAIDS EVEN IF APPLE DOESN'T!!!!

G5 XSERVE CLUSTERNODES $1429
G5 XSERVES FULL UNITS $1849!!!!
1 TB XSERVE RAIDS FROM $4299!!!!
2.8 TB XSERVE RAIDS FROM $4999!!!!
5.6 TB XSERVE RAIDS FROM $6299!!!!
3.5 TB/7.0 TB XSERVE RAIDS $5899/$7999
RAID CARDS, FIBRE CARDS, DRIVE MODULES
POWER SUPPLIES, CONTROLLER MODULES ETC
OVERNIGHT SERVICE AVAILABLE!!!!

Displays

REFURBISHED DISPLAYS
17" Studio LCD: $299
22"/23" Cinema: $499/$599
15" Studio LCD: $99
17" Studio CRT,ADC/VGA: $49.99

Products refurbished or demo, call for more information

method of mutable arrays is `addObject:` that adds an object to the end of the array:

```
NSMutableArray * animals = [NSMutableArray array];
[animals addObject: @"cat"];
[animals addObject: @"dog"];
```

Again, there are far more methods to `NSArray` and `NSMutableArray` to cover here, but you've learned enough to get you started.

Dictionaries

Another popular collection is the *dictionary*. Dictionaries manage pairs of keys and values. Dictionaries can also efficiently look up values by their keys. In other languages, dictionaries are known as *hash tables* or *associative arrays*. You can think of a dictionary as a lookup table. For example, let's say we have a table of countries and their capitol city:

Table 1: Countries and their capitals

Country	Capitol
United States	Washington, D.C.
England	London
France	Paris

Let's say we wanted to use the information in Table 1 to create a lookup table so we could quickly find the capital of a country. We could use a dictionary to do this. Each row in the table is a key/value pair. The country is the key, since that is what we are using as the lookup, and the capitol is the value.

The Foundation framework has a class called `NSDictionary` that is a dictionary implementation, with one minor change: it calls values "objects". We can create a new dictionary with the `dictionaryWithObjectsAndKeys:` class method, set the countries to be the keys, and set the capitals to be the values, or objects, as `NSDictionary` likes to call them:

```
NSDictionary * capitals =
    [NSDictionary dictionaryWithObjectsAndKeys:
     @"Washington, D.C.", @"United States",
     @"London", @"England",
     @"Paris", @"France",
     nil];
```

Note again the key/value pair list is terminated with a `nil`. With this dictionary, we can now look up a capital (object) given a country (key) using the `objectForKey:` method:

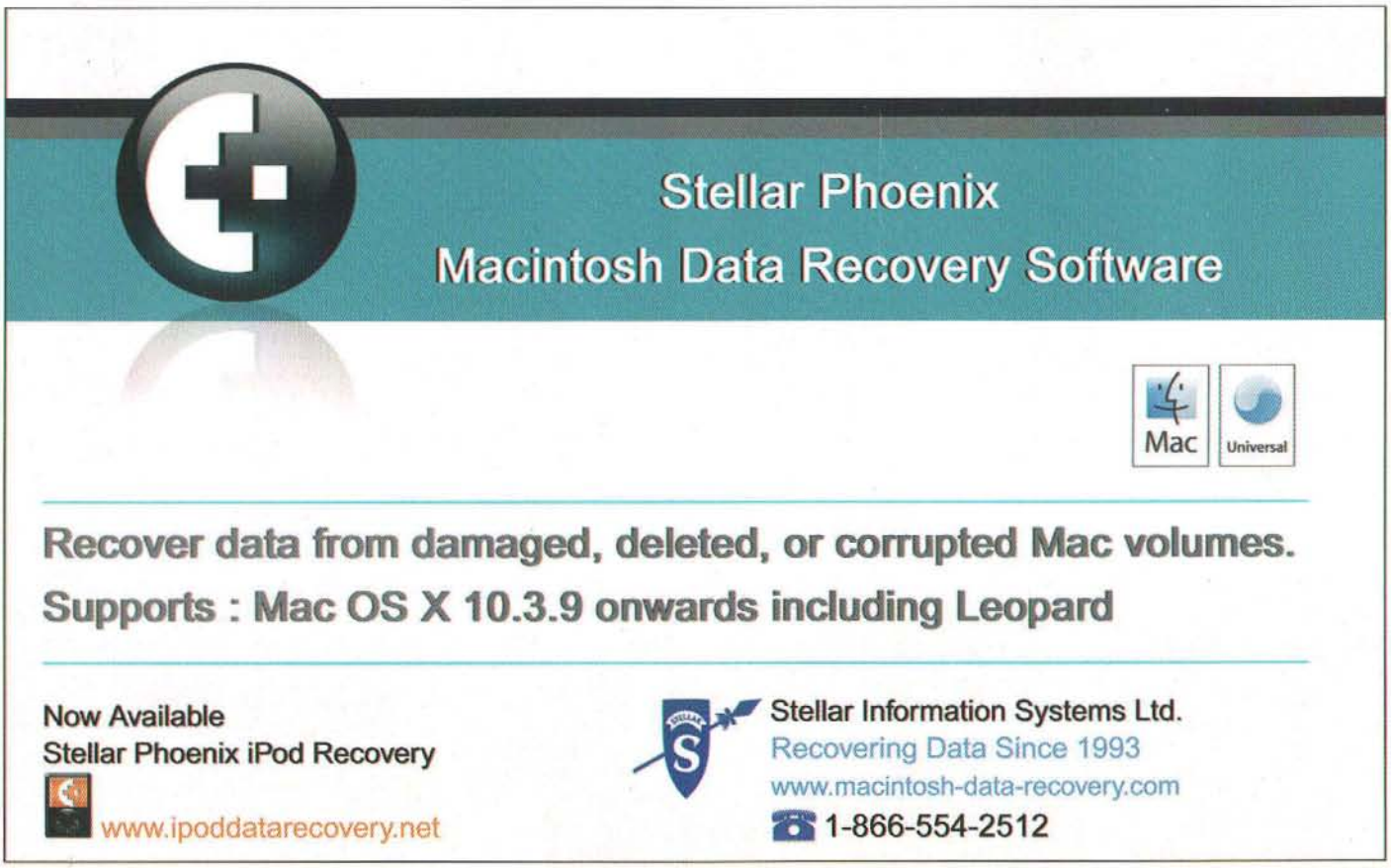
```
    NSString * capital = [capitals objectForKey:
@"England"];
    printf("Capital of England is %s\n", [capitol
UTF8String]);
```

If the key has no corresponding object, then it returns `nil`.

As with arrays and strings, `NSDictionary` is immutable. If you want an updateable dictionary, use the `NSMutableDictionary` subclass.

Loose Ends

All collection classes retain their objects. This means that after you add an object to an array or dictionary, you may



release your instance of it, if you no longer need it. The collection classes will release an object when it is removed from the collection, or they will release all their objects when they themselves get deallocated.

One downside to the collection classes in Foundation is that they can only hold Objective-C objects. This means that you cannot put standard C types, or *primitive types*, such as `int` and `float`, directly into a collection. Luckily, Objective-C has wrapper classes for primitive types, one of them being `NSNumber`. `NSNumber` is an immutable class that holds any primitive number type. Here's how you would put the integer 42 into an `NSNumber`, and then get it back out again:

```
NSNumber * theAnswer = [NSNumber numberWithInt: 42];
printf("The answer is %d\n", [theAnswer intValue]);
```

Since `NSNumber` is a full-blown Objective-C class, you can use it to put primitive numbers into arrays and dictionaries. I will demonstrate this shortly.

Election Counting

Putting together everything we've learned in this article, we're going to write a small application that tallies votes for an election. This seems rather fitting with 2008 being an election year in the United States. However, instead of voting for president, let's take a poll that asks people to vote on their favorite fruit. Since counting up votes can be tedious, we'd like to write an application to tally up the results. Using the classes in Foundation, this is actually pretty easy! Listing 1 shows the entire program. Read it over quickly, and then we'll walk through it.

Listing 1: tally.m: A vote tallying program

```
#import <Foundation/Foundation.h>

int main (int argc, const char * argv[])
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc]
init];

    NSString * results =
        @"apple,orange,apple,cherry,banana,apple,banana,"
        @"orange,apple,banana,cherry,banana,apple,orange";

    NSArray * votes = [results componentsSeparatedByString:
@","];

    // Tally up the votes
    NSMutableDictionary * tallies = [NSMutableDictionary
dictionary];
    int i;
    for (i = 0; i < [votes count]; i++)
    {
        NSString * vote = [votes objectAtIndex: i];
        NSNumber * currentTally = [tallies objectForKey:
vote];
        int newTally;
        if (currentTally == nil)
```

```
        {
            // This is the first vote for this candidate
            newTally = 1;
        }
        else
        {
            newTally = [currentTally intValue] + 1;
        }

        [tallies setObject: [NSNumber numberWithInt:
newTally]
                    forKey: vote];
    }

    // Print out the results
    int winningTally = 0;
    NSString * winner;
    NSArray * voteKeys = [tallies allKeys];
    for (i = 0; i < [voteKeys count]; i++)
    {
        NSString * vote = [voteKeys objectAtIndex: i];
        NSNumber * tally = [tallies objectForKey: vote];
        printf("%10s: %d\n", [vote UTF8String],
               [tally intValue]);
        if ([tally intValue] > winningTally)
        {
            winner = vote;
            winningTally = [tally intValue];
        }
    }
    printf("\nAnd the winner is: %s!\n",
           [winner UTF8String]);

    [pool release];
    return 0;
}
```

The general idea of this program is to take a comma-separated list of votes and tally them up. Once we have the tallies, we print the results and then the winner.

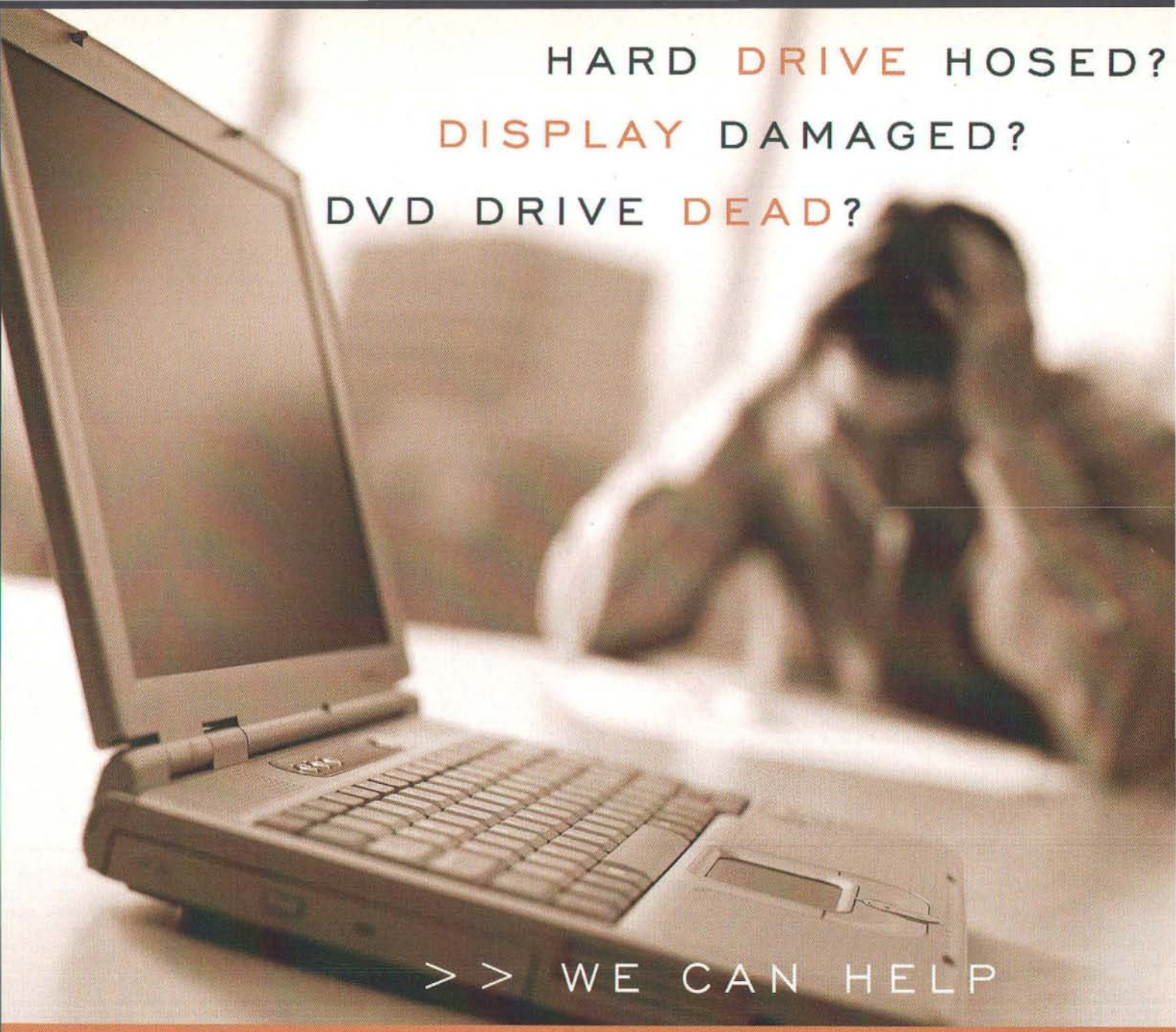
The first new syntax you'll see is how the `results` string is created. This shows how you can break up long string literals onto multiple lines. If you don't end the first line with a comma or semicolon, you can just start the second line like a normal string. The result is one big `NSString`.

The next new bit is how we split the results into individual votes. We use the `componentsSeparatedByString:` method to split the long, comma-separated string into an array of strings. It also removes the commas, so we have a nice, clean array of votes.

With each vote now in an array, we can proceed to calculating the tallies. We use a mutable dictionary to tally up the votes, where the key is the vote and the value is the current tally. The only complication is that we have to use the immutable `NSNumber` class to store the tally, since primitive types cannot be stored in a dictionary.

We loop through each vote in the `votes` array and look up its current tally in the `tallies` dictionary. Since initially the dictionary is empty, we may not get any tally back. If this is the case, then `objectForKey:` will return `nil`. We use this condition to set the next tally to 1. Otherwise, we add 1 to the current tally. We then package up the new tally as an `NSNumber` and store it back in the dictionary with the `setObject:forKey:` method. This method replaces any existing value, so our dictionary will only contain the current tally.

After we are done looping through all the votes, the `tallies` dictionary contains our voting results. Now we need to report the final results. The easiest way to do this is to loop through every key/value pair in the dictionary, and the easiest way to do *that* is to use the `allKeys` method of



REPLACEMENT PARTS • UPGRADES • REPAIRS FOR YOUR MAC LAPTOP

- FREE ONLINE DO-IT-YOURSELF GUIDES
- SEND-IN REPAIR SERVICE
- BATTERY REPLACEMENTS FROM $ 94.95
- POWER ADAPTERS FROM $ 24.95

VISIT US ON THE WEB>>
OR CALL 866.726.3342

PowerbookMedic.com

Your Mac – Our Patient

w.PowerbookMedic.com
5 Sparkman Dr. Ste. 1620
ntsville, AL 35816
/Fax: 866.726.3342

`NSDictionary`. It gives us every key in an array. We can then loop through each key, get its corresponding value, and print the tally. In order to make the results line up in columns, we use the `%10s` format specification for `vote`. This tells `printf` to pad out the string to 10 characters using spaces, if the string is less than 10 characters.

As we are looping through all the tallies to print them out, we also keep track of the maximum number of votes so we can find our winner. After reporting the final results, we print out our winner.

Okay, so what happens when we run this application? I get the following output:

```
banana: 4
cherry: 2
orange: 3
 apple: 5

And the winner is: apple!
```

It was a close race, but ultimately apple (or is that Apple?) prevails. Well, as cool as this program is, it does have some limitations. First, the `results` string is stored directly in the program. This means if we need to update our results, we also have to recompile our program. Ideally, we would store our results in an external text file, but that would complicate this simple example a bit too much. If you want to try this yourself, though, look into the `stringWithContentsOfFile:encoding:error:` method of `NSString`. Here's a little hint to get you started. It reads a file named `results.txt` on your desktop into a string:

```
NSString * file = @"~/Desktop/results.txt";
file = [file stringByExpandingTildeInPath];
NSString * results =
    [NSString stringWithContentsOfFile: file
                              encoding:
NSUTF8StringEncoding
                                 error: NULL];
```

Be careful, though. Even this code is not complete, as it ignores any errors that may occur. Production code should *always* handle errors. We will talk more about properly handling `NSError` later. If you want to try this out anyway, an added challenge would be to support votes on separate lines, instead of separated by commas.

Another limitation is that the results are not printed in any order. Ideally, we would print the results in ascending or descending order, by votes. The `allKeys` method does not guarantee what order the keys are in. In fact the order could be different *every time* we run it. Getting the keys back in a specific order requires some topics we haven't yet covered, such as selectors and comparators. If you want to learn more about this on your own, look into the `keysSortedByValueUsingSelector:` method of `NSDictionary`.

The final limitation is that we don't handle ties. We could use an array of winners, instead of a single winner, to fix this. This would be another fun modification to try on your own.

Conclusion

Well, we are making good progress! We've learned about the string, array, and dictionary classes that come as part of the Foundation framework. There's much more to Foundation, but with even this basic knowledge, we can do a lot. In fact, we can even begin to write GUI applications. Thus finally, next month, we will step away from the dark world of text-only command line applications, and start writing GUI applications.

Be smart - Backup different

SmartBackup

The alternative lightweight backup solution.

"Using Saved Searches as include and exclude rules is just brilliant..." Macgeekery.com

"Extremely simple - surprisingly powerful" Macapper.com

TRY NOW! Trial version available at: http://freeridecoding.com/smartbackup

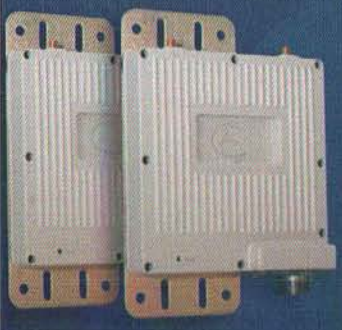
About The Author

Dave Dribin has been writing professional software for over eleven years. After five years programming embedded C in the telecom industry and a brief stint riding the Internet bubble, he decided to venture out on his own. Since 2001, he has been providing independent consulting services, and in 2006, he founded Bit Maki, Inc. Find out more at <http://www.bitmaki.com/> and <http://www.dribin.org/dave/>.

WE OUTPERFORM OUR COMPETITION

Nobody beats us on...

- **Price**
  Save thousands on multi-location T1 services
- **Performance**
  6 to 30× faster than T1 service
- **Reliability**
  Best-in-class fixed wireless solutions



Just one of many wireless solutions from Trango...

ATLAS 5010 Series™
Wireless Ethernet Bridge

- 45 Mbps of sustained throughput
- Capable of 5.3 and 5.8 GHz
- Point-to-point, OFDM
- Up to 40-mile range
- Adaptable Rate Modulation

Wireless Connectivity IS Powered by Trango

Trango Broadband Wireless
A division of Trango Systems, Inc.

www.trangobroadband.com
Phone: +1 (858) 653-3900

trangobroadband WIRELESS

MACTECH LABS

Office 2008 Benchmarks

How well does Office 2008 run compared to Office 2004?

by Neil Ticktin, Editor-in-Chief

The Big Question

If you are a Microsoft Office user on the Mac, there's likely a question on your mind about the new version of Microsoft Office 2008 for Mac. As you may remember, MacTech did extensive benchmarking on Microsoft Office 2004 for Mac, running under Rosetta, when the Intel Macs first came out. See http://www.mactech.com/articles/mactech/Vol.22/22.05/Office2004Benchmark/ to read the full article.

But Office 2008 is "Universal," meaning that it's designed to take advantage of the Intel processor, while still being compatible with PowerPC based machines. The big question is therefore, "How fast is the new Office 2008?"

To answer that question, we put Office 2008 through its paces on both Tiger and Leopard (Mac OS X 10.4.11 and Mac OS X 10.5.1 to be specific). With over 2,500 tests, we looked at Word, Excel, PowerPoint and Entourage as well as the graphics library resources that are shared across all of Microsoft Office for Mac.

Overview

We won't keep you in suspense. In general, Office 2008 is faster than Office 2004 when run on Intel Macs. On the PowerPC (represented by our PowerBook G4), it certainly runs "well enough" albeit marginally slower.

In the over 2,500 real world tests comparing Office 2004 with 2008, the vast majority were faster, with many features being 2-3x faster in the new version. On average, Office 2008 running on Intel was 28% faster than 2004 for those items we tested.

For PowerPC users, the issue of whether to upgrade should come down to whether you want the new features, or new file format supported integrated into the application.

Clearly, however, for Intel users, there are not only the benefits of new features and a more straightforward user interface, but many speed improvements as well.

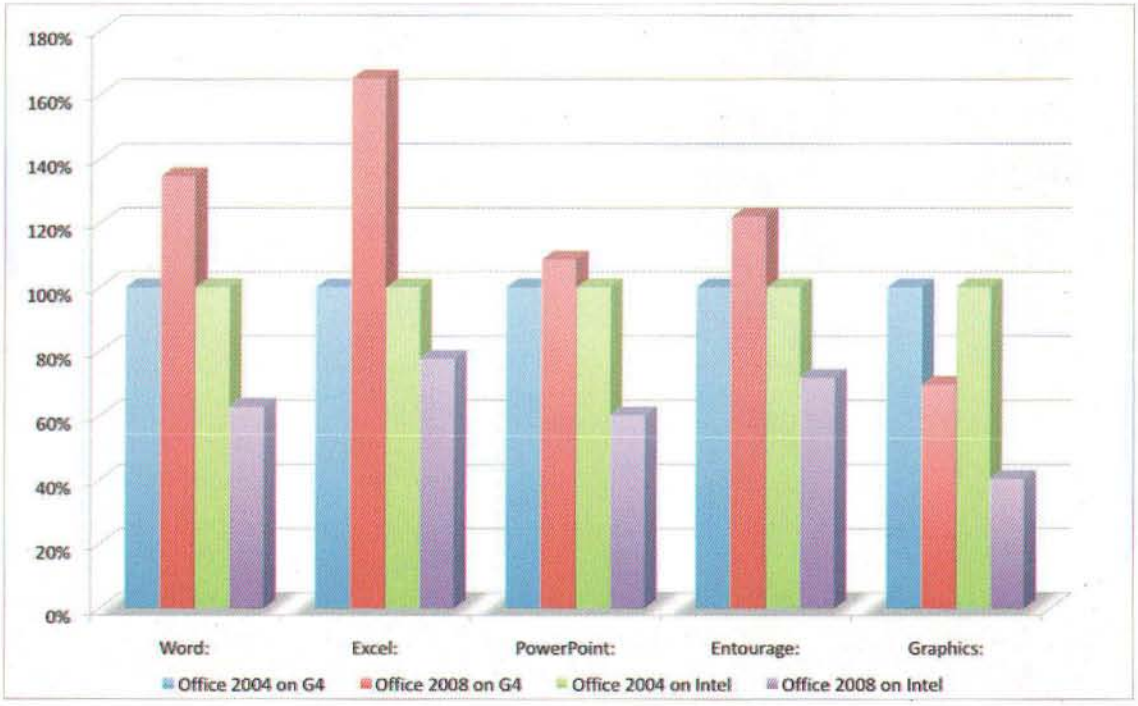


Figure 1: Office 2008 vs. Office 2004? (shorter is faster)

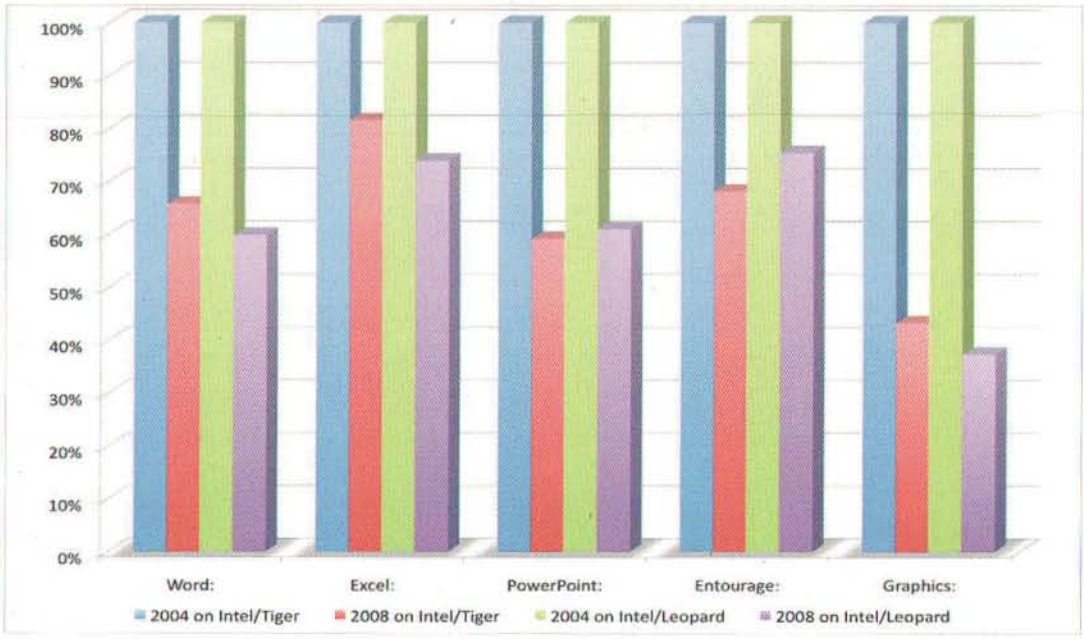


Figure 2: Intel Tiger vs. Intel Leopard? (shorter is faster)

In Figure 1, you can see how Office 2008 ran compared to Office 2004. The PowerPC machines (blue and red) and Intel machines (green and purple) represent show the two different setups for Office 2004 and 2008, respectively.

To determine this, MacTech ran tests across five models of Macs: PowerBook G4, MacBook, MacBook Pro, Intel iMac, and Mac Pro. Real world tests were run to make these determinations, such as:

successive launch
scrolling
find and replace
opening files
pasting
printing
zooming
subtotals
auto formatting
applying templates
IMAP account sync
Searching
Inserting and importing graphics

Furthermore, for most users when running Office 2008 on Intel, it's usually slightly faster on Leopard than it is on Tiger. The only exceptions were PowerPoint and Entourage which ran just a bit slower. See Figure 2.

Application Overall Results

We ran tests in each of the four main Office 2008 apps, plus the graphics engine. What we found for each of them was as follows.

For Intel Mac users, on average:

- Word 2008 ran 37% faster than Word 2004
- Excel 2008 ran 22% faster than Excel 2004
- PowerPoint 2008 ran 40% faster than PowerPoint 2004
- Entourage 2008 ran 28% faster than Entourage 2004
- Graphics (common to all the applications) in Office 2008 ran 60% faster than Office 2004

To make this even more clear, when something says "30% faster," that means that if it took 10 seconds on 2004, it took 7 seconds on 2008. This tends to understate "faster" when compared to perception.

While we already had thousands of tests to perform, we thought it important to include a PowerPC machine in the mix. We chose the PowerBook G4 as a reasonable representation of PowerPC. Obviously, our tests needed to focus on machines shipping in the last couple of years, and that's Intel based Macs. But, if you are looking to deploy 2008 on PowerPC based Macs, either because you want consistency, the new file formats, or the new features, 2008's speed on PowerPC is very usable, and the slower stopwatch tests shouldn't stop you.

For PowerPC Mac users, represented by our PowerBook G4 testing, on average:

- Word 2008 ran 35% slower than Word 2004
- Excel 2008 ran 65% slower than Excel 2004
- PowerPoint 2008 ran 9% slower than PowerPoint 2004
- Entourage 2008 ran 22% slower than Entourage 2004
- Graphics in Office 2008 ran 30% faster than Office 2004

Test Environment

When we were choosing computer models, we set out to choose not the fastest, latest models, but ones that would be a good representation of what most people may have. Certainly, the faster models of these computers will perform even better.

Specifically, these are the machines that we used:

- PowerBook G4: 2GB RAM, 1.5 GHz G4 processor
- MacBook: 2GB RAM, 1.83 GHz Core Duo processor
- MacBook Pro: 4GB RAM, 2.16GHz Core 2 Duo processor
- iMac: 2GB RAM, 1.83 GHz Core 2 Duo processor
- Mac Pro: 4GB RAM, Quad Core (two 2.66 GHz Dual-Core Intel Xeon processors)

The test bench included configurations of both Mac OS X 10.4.11 and Mac OS X 10.5.1, with Office 2004 and 2008 installed. All installations were completely clean installations of both Mac OS X and Microsoft Office.

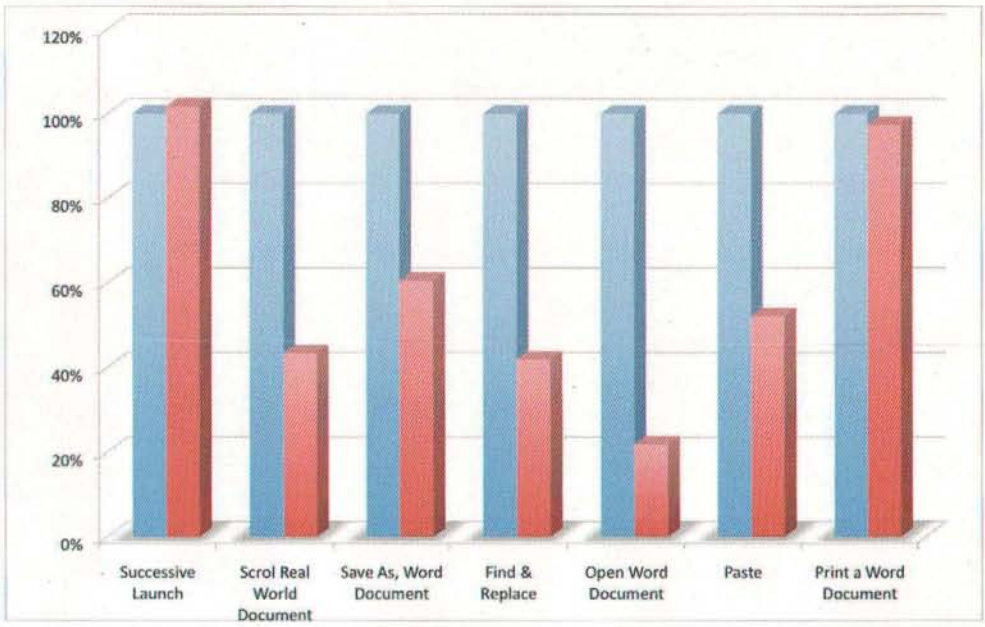


Figure 3: Word 2004 vs. Word 2008? (shorter is faster)

Specific Applications and Tests

The number of tests that we did across the number of Mac models across two Mac OS X versions would make looking at charts and results too complex for any normal human. With that in mind, let's focus on Office 2008 vs. Office 2004 on

Mobile Electronics Repairs & Upgrades
Overnight - Nationwide
• Local Pickup & Delivery
• Fast Friendly Service
• 2000+ Local Drop-Off Locations

1-888-64-RESTORE
(1-888-647-3786)
8am - 5pm PST M - F
techrestore.com

Put A Smile On Your Client's Face
By Referring Them To The Nation's
Most Recommended Repair Service

Laptops • iPods • PSPs • Apple TVs
Repairs/Upgrades/Data Recovery

- Affiliate Commissions
- Reseller Accounts
- Volume Parts Sales
- Blind Drop Shipments
- Transparent Back-End Repairs

Outsource Repairs,
Increase Profits!

www.techrestore.com/reseller

America's Most Trusted Repair Service Now Offers Over 2000 Local Drop Off Locations With America's Most Trusted Carrier

FedEx Kinko's
Office and Print Center

Shop for Mac Laptops, Apple TVs & Mobile Electronics Accessories

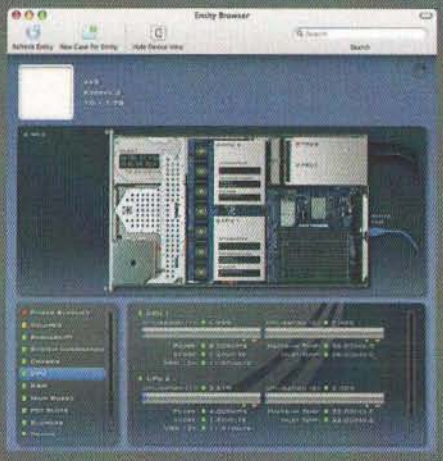
Tech e-store

FedEx service marks used by permission

Network, Server and Appliance Monitoring.

For Mac OS X

Xserve RAID



Xserve (Intel & G5)

Airport

Lithium Network Monitoring Platform

Lithium can now monitor your Xserve, Xserve RAID, Qlogic switches, Airports, Mac OS X Server... and everything else in your network.

www.lithiumcorp.com

lithium Corp.

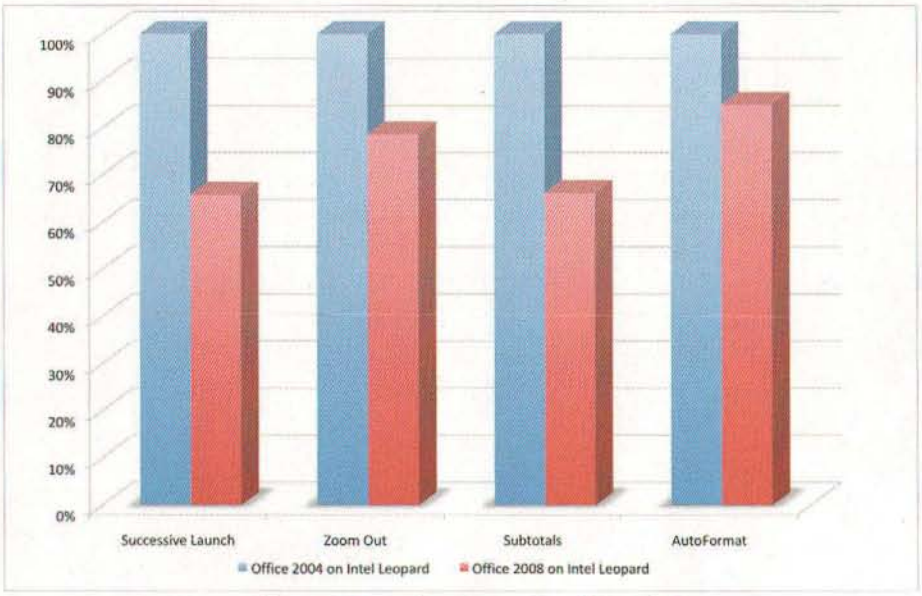


Figure 4: Excel 2004 vs. Excel 2008? (shorter is faster)

Leopard on Intel machines as this is the most common scenario. (Don't worry, the complete results for individual machines and tests are available in the results tables if you want to see them.)

Here is what we found for each application (Word, Excel, Entourage and PowerPoint). Note: Both Office 2004 and Office 2008 use a graphics engine that is shared by all the Office applications. With that in mind, the final set of tests focuses on that graphics engine.

Microsoft Word: Specifics

Clearly, Word is the most used application in the Office Suite. As a result, it makes sense for us to run the largest test suite in Word. The test suite included:

- Successive launch (i.e., a launch after the application had been launched already once)
- Scrolling from top to bottom in a document
- Save as (not "save") of a real world document that included text, pictures, and more.
- Global find and replace
- Opening a larger document
- Pasting
- Printing a document with complexity (e.g., text, pictures, etc…)

As you can see from Figure 3, Word 2008 is a whole lot faster than Word 2004 with the exception of launching (slightly slower) and printing (which relies more on the OS).

The documents used for the test were real world sized, but on the larger side so that we could get quality test results.

Is your business drowning in e-mail?

MailTank is your solution. Try it free.

Learn more at **http://mailtank.com** or call **877-622-8265**

HASP, the #1 Choice of Software Developers Worldwide*

*Source: Frost & Sullivan #N1AF-70, IDC Bulletin #34452

FROST & SULLIVAN

Finally, put a

STOP

to unauthorized software use.

HASP® HL

REINVENTING SOFTWARE PROTECTION & LICENSING

Follow the signs to HASP HL – the market's leading software DRM solution

HASP HL provides the highest level of security for your software. Featuring easy-to-use tools, HASP HL automatically protects your software and enables new and innovative licensing options.

- ***Strong copy protection:*** *Powerful 128-bit AES encryption provides a strong locking mechanism that ensures you get paid for every copy of your software in use.*
- ***Unmatched intellectual property protection:*** *Automatically protects Windows executables, DLLs, Mac applications, and offers class-level obfuscation for .NET framework 2.0 applications. Multiple encryption layers ensure your valuable IP remains unreachable and your R&D investment is protected.*
- ***Secure & flexible licensing:*** *HASP HL enables a variety of secure and flexible licensing models and ensures end-users comply with software licensing terms.*

See for yourself why IDC ranks HASP #1.

Request a Software Developer Kit today at www.Aladdin.com/StopPiracy_MT

Aladdin®

SECURING THE GLOBAL VILLAGE

• NORTH AMERICA: 1-800-562-2543, 847-818-3800
• UK • GERMANY • ISRAEL • BENELUX • FRANCE • SPAIN • ITALY • BRAZIL • INDIA • CHINA • JAPAN

© Aladdin Knowledge Systems, Ltd. All rights reserved. Aladdin and HASP are registered trademarks of Aladdin Knowledge Systems, Ltd.

www.Aladdin.com/HASP

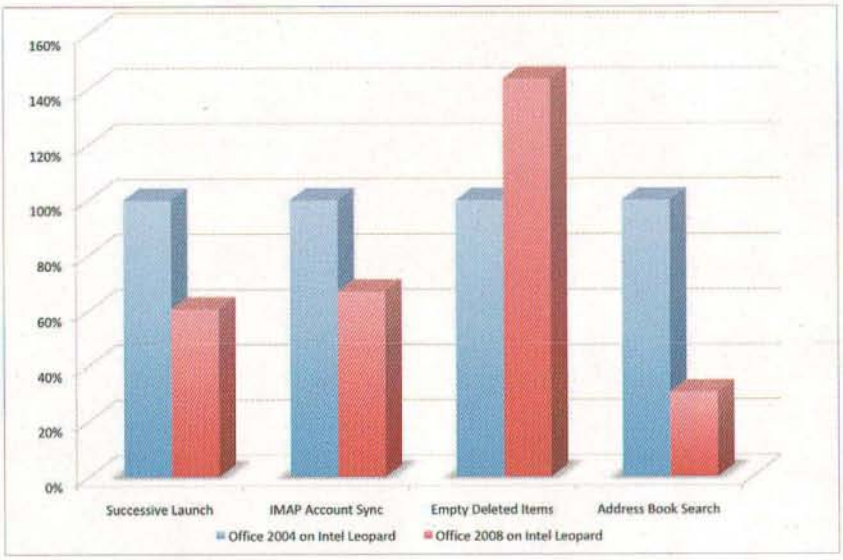


Figure 5: Entourage 2004 vs. Entourage 2008? (shorter is faster)

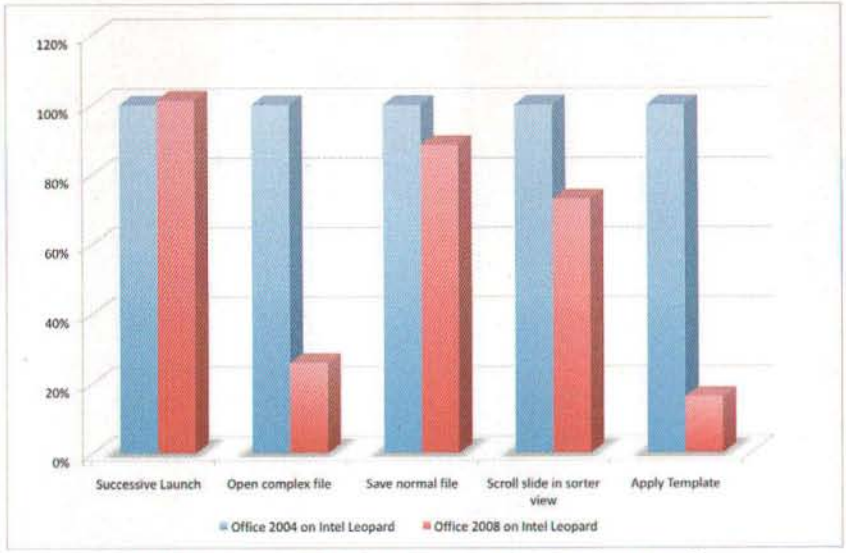


Figure 6: PowerPoint 2004 vs. PowerPoint 2008? (shorter is faster)

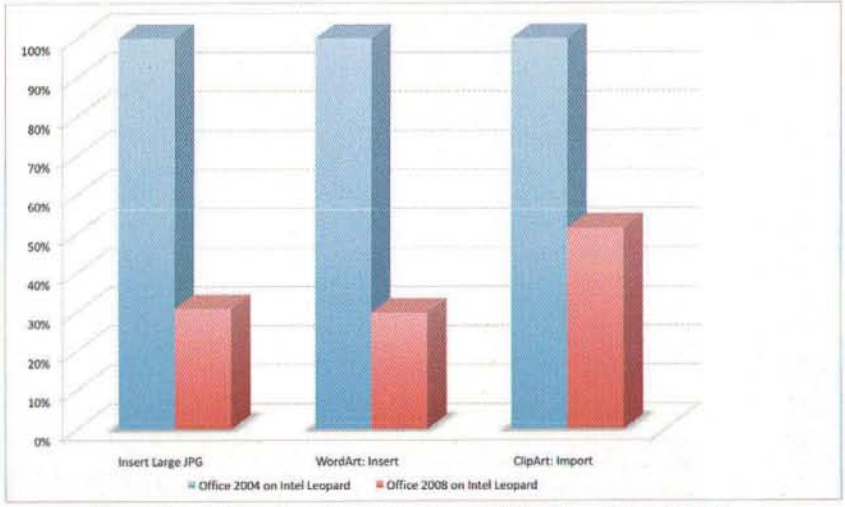


Figure 7: Office Graphics 2004 vs. Office Graphics 2008? (shorter is faster)

Microsoft Excel: Specifics

For Excel, we focused on things that the application spent time "thinking about." As a result, the test suite included:

- Successive launch
- Zoom out
- Subtotals (and the formatting that goes along with that)
- Auto format

We also looked at scrolling, but it turns out that scrolling was so fast that Microsoft's user testing showed it needed to be slowed down. As a result, the test wasn't appropriate.

In all cases, Excel 2008 was faster than Excel 2004.

Microsoft Entourage: Specifics

For Entourage, we focused on the items that were measurable. As a result, the test suite included:

- Successive launch
- IMAP account sync
- Purge deleted items
- Address book search

We had also looked at POP message download and sending messages via SMTP, but found that these were not good tests of the application … that they had more to do with the network and the server. Additionally, we looked at mail and task search, but found the tests ran too fast to be measured in Entourage 2008.

Microsoft PowerPoint: Specifics

So much of one's time in PowerPoint is spent reviewing slides and transitions, as well as typing. These are not good tests, however, as they test either the user, or are fixed length by design (e.g., a transition speed doesn't typically vary much). As a result, the test suite included:

- Successive launch
- Opening a complex PowerPoint file
- Saving a normal size file
- Scrolling through a large presentation in "sorter view"
- Applying a template throughout a large presentation

In all cases except one, PowerPoint was faster. Sometimes much faster. Launching was slightly slower.

Microsoft Office Graphics: Specifics

As we mentioned earlier, Office 2008 and 2004 both rely on a common graphics library that is

OWC® Mercury Rack Pro™ 4-Bay 1U RackMount Storage Solution

Ideal for applications requiring high data throughput, availability, and flexibility in configuration.

1.2TB to 4.0TB from $689.99

- Hardware RAID, SoftRaid, JBOD options.
- Available with eSATA, FireWire 800/400, and USB 2.0 interface support.
- User customizable configurations.
- Up to 4.0TB of performance storage.

OWC® Mercury Elite™-AL Pro Storage Solution

All Mercury Elite-AL Pro models are ideally configured for Audio, Video, Digital Photography, Professional Music, Graphics, General Data and Back-Up applications.

250GB to 1.0TB from $139.99

- Rugged, machined aluminum enclosure & super quiet operation.
- Up to 32MB data buffer with data transfer rates over 150MBs!
- Ultra-protective shock isolation system.
- Premiere BONUS utility software included.
- Available with eSATA, FireWire 800/400, and USB 2.0 interface support.

OWC® Mercury On-The-Go™ Bus Powered Portable Solution

Safely and conveniently transport large amounts of data with no AC adapter needed!

80GB to 320GB from $89.99

- FW 800/400/USB2.0 to USB2.0/eSATA solutions.
- Fully suitable for audio/video applications.
- Super quiet operation with shock isolation system.
- Premiere BONUS utility software included.
- Compact 3.5"(W) x 5.5"(D) x 1"(H) size and weighs less than 11 ounces. Fits in your shirt pocket!

Other World Computing™

Visit: www.owcmac.com or Call: 800.275.4576

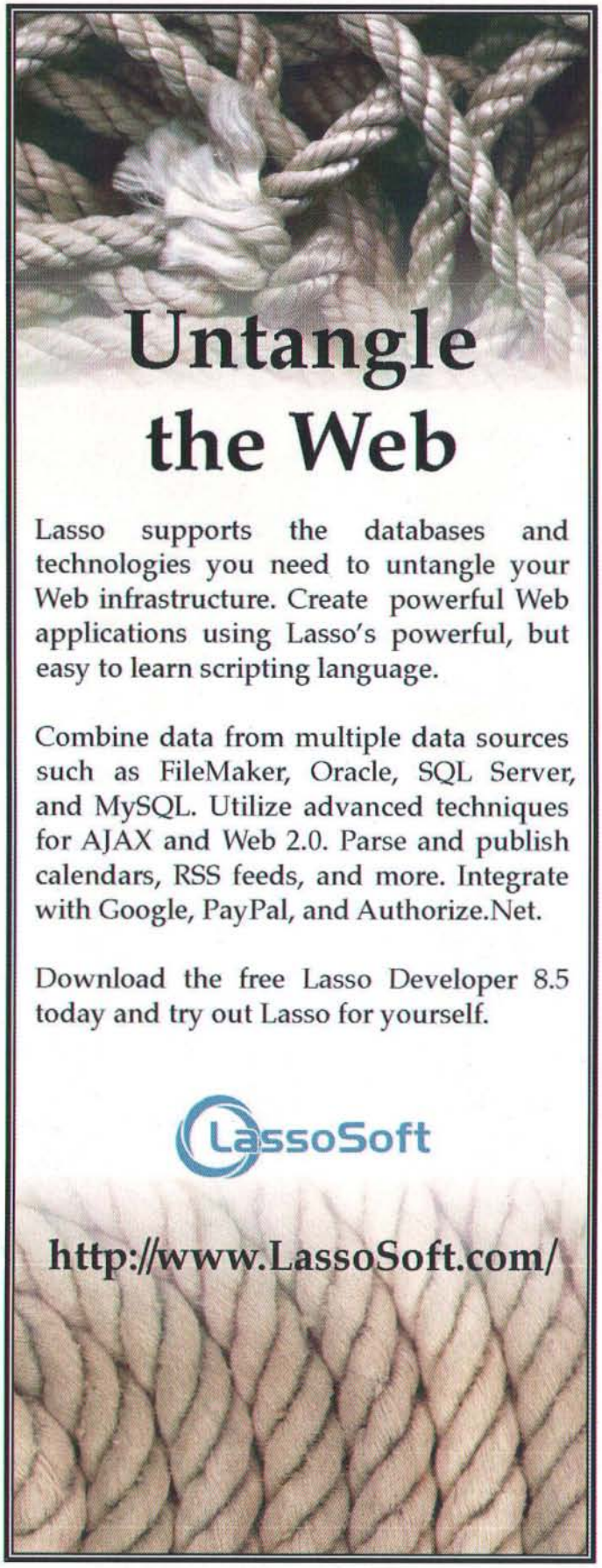
Award-winning products: Macworld, MacStuff, LAYERS, Photoshop User, Extreme Drive Appraisal, Hard Mac Guide

TIME MACHINE READY

Time Machine ready means that our external hard drives keep an up-to-date copy of all of your files and document under OS X Leopard™. Go back in time to recover anything with ease.

BizRate 2007

Mercury Elite, Mercury On-The-Go, Mercury Rack Pro, and Other World Computing are trademarks, OWC and OWC logo are registered trademark of Other World Computing. Other marks may be the trademark or registered trademark property of their owners. Prices, specifications, and availability are subject to change without notice.



used across the Office suite. With that in mind, we tested some of the graphics functions to see how they fared, including:

- Inserting a large JPG
- Insertion WordArt
- mporting a number of pictures into ClipArt

In all of these cases, Office 2008 graphics were substantially faster.

Some Speed You Can't Stopwatch

MacTech's tests are primarily done using a stopwatch, and as such, they are timing things that are done solely by the computer. This is a good judge of the difference between Office 2004 running under Rosetta vs. Office 2008 running as a Universal application, native on Intel.

But, what Office 2008 does more than just be Universal is to bring some concrete advancements in creating work, as well as the level of quality that a "normal" person can accomplish. This is done through Document Elements, SmartArt, Ledger Sheets, Charts, and other devices. Clearly these are strongest in Word and Excel.

What these features do for the human being is make it easier to get there — we'll call this "user speed." For example, charts can be created in one click, without having to go through dialog box after dialog box in a wizard. Or, building a formula in Excel. It's far easier than it has been in a long time.

From a quality of work product, Word and Excel in particular have new features that really raise the bar. For example, the templates provided vastly enhance the quality of a "newsletter" or a "flyer". And, SmartArt: wow. I can't even imagine any student or business user even thinking of doing this in any Office product before.

All of these items give users a great deal more "user speed" ... and those are not the type of things that can be measured with a stopwatch. But, in many cases, you will see significant "user speed" improvements, regardless.

Conclusion

As we stated at the beginning of this article, on Intel machines, Office 2008 runs faster for most things than 2004 does. For PowerPC based machines, 2008 works "well enough"; in fact, for many things, we don't think you'll notice much of a speed difference from 2004 on PowerPC.

For those that work in areas affected by the "user speed" enhancements, and we have to believe that's the majority of users, you'll see substantive improvements. Add to it the new features, and you have an application suite that allows you produce higher quality work, in far less time, either because it's Universal and faster, or because of what the new user interface and templates bring you.

In the end, if you want the new features, or the new file formats, then Office 2008 is the way to go.

Table: Mac Model/OS Specifics When Comparing Office 2008 to Office 2004

Office 2004 = 100% Baseline. Lower % Values mean Faster

	PowerBook G4 on 10.4.11	PowerBook G4 on 10.5.1	MacBook on 10.4.11	MacBook on 10.5.1	MacBook Pro on 10.4.11	MacBook Pro on 10.5.1	iMac on 10.4.11	iMac on 10.5.1	Mac Pro on 10.4.11	Mac Pro on 10.5.1
Word:										
Successive Launch	153%	305%	387%	103%	103%	118%	109%	106%	90%	105%
Scroll Real World Doc	61%	108%	134%	45%	37%	57%	51%	45%	42%	47%
Save As Word Doc (normal)	78%	108%	108%	86%	38%	81%	64%	78%	63%	75%
Find & Replace	57%	114%	110%	43%	42%	46%	42%	44%	40%	46%
Open Word Doc	27%	42%	38%	31%	29%	32%	28%	25%	18%	18%
Paste	66%	117%	120%	52%	50%	57%	56%	53%	50%	51%
Print a Word Doc	98%	99%	94%	103%	96%	96%	95%	95%	100%	100%
Excel:										
Successive Launch	99%	225%	235%	59%	56%	74%	72%	68%	63%	68%
Zoom Out	107%	194%	204%	91%	80%	82%	73%	89%	81%	93%
Subtotals	79%	109%	97%	61%	45%	89%	87%	68%	42%	100%
AutoFormat	96%	133%	123%	100%	88%	92%	87%	79%	80%	91%
PowerPoint:										
Successive Launch	103%	119%	206%	77%	100%	77%	112%	74%	93%	73%
Open complex file	39%	83%	110%	23%	25%	24%	27%	25%	26%	24%
Save normal file	105%	127%	133%	109%	89%	110%	93%	110%	82%	108%
Scroll slide in sorter view	78%	103%	100%	60%	58%	63%	70%	73%	72%	89%
Apply Template	23%	48%	57%	15%	14%	17%	18%	17%	17%	16%
Entourage:										
Successive Launch	90%	189%	192%	67%	59%	72%	68%	65%	50%	67%
IMAP Account Sync	74%	95%	123%	65%	65%	61%	67%	62%	65%	63%
Empty Deleted Items	132%	147%	155%	116%	143%	91%	114%	116%	167%	118%
Address Book Search	32%	38%	38%	35%	36%	25%	28%	33%	26%	32%
Graphics:										
Insert Large JPG	34%	47%	39%	22%	20%	40%	38%	32%	30%	35%
WordArt: Insert	40%	81%	62%	37%	31%	42%	29%	30%	30%	29%
ClipArt: Import	65%	92%	96%	53%	44%	67%	59%	60%	53%	72%

About The Author

Neil is the Editor-in-Chief and Publisher of MacTech Magazine. Neil has been in the Mac industry since 1985, has developed software, written documentation, been heading up the magazine since 1992. When Neil writes a review, he likes to put solutions into a real-life scenario and then write about that experience from the user point of view. That said, Neil has a reputation around the office for pushing software to its limits and crashing software/finding bugs. Drop him a line at publisher@mactech.com

What's in Your Target

Unit testing and analysis coverage

by Aaron Montgomery
with Dave Dribin,
contributing editor

Introduction

If you are building projects with Xcode, you are already using targets in your project. The target collects together information about how to build a library or application. If you are working with more complicated projects, you may have one target that builds a library and a second target that builds an application that depends on that library. This article describes Xcode targets that help in auxiliary tasks. Using an Xcode target to produce documentation has been discussed in MacTech (see the references at the end of the article). In this article, we present a target that runs unit tests using the CPlusTest framework for Carbon applications (there is also a Sen Testing Kit for Cocoa applications, but we will not cover that here). We will then add a shell script that allows us to use the Linux Coverage Tool to analyze how much of our code we are executing. The inspiration for this article was the November 3 2005 entry in Chris Liscio's log that discussed how to add `gcov` analysis to unit testing (see the references). This article assumes you are working with Xcode 3 and building for a Mac OS X 10.5 target. I have done similar projects for Xcode 2.2 and 2.3 on Mac OS X 10.4 and will point out differences for those configurations as we go along.

We start with a simple application called SuperAdd that implements a "highly optimized" adding routine. The application started as a basic Carbon Application project and we will assume that the reader already has the skills required to create a Carbon application with Xcode.

Unit Tests

Before I discuss how to add a testing target, a few words are in line about what unit tests can do, and (more importantly) what they cannot do. Unit tests are designed to call your functions with inputs that you specify and then verify that the function produced the correct output. Unit tests do not debug your code. They may help you determine which section of code is problematic, but they cannot tell you how to fix the problem.

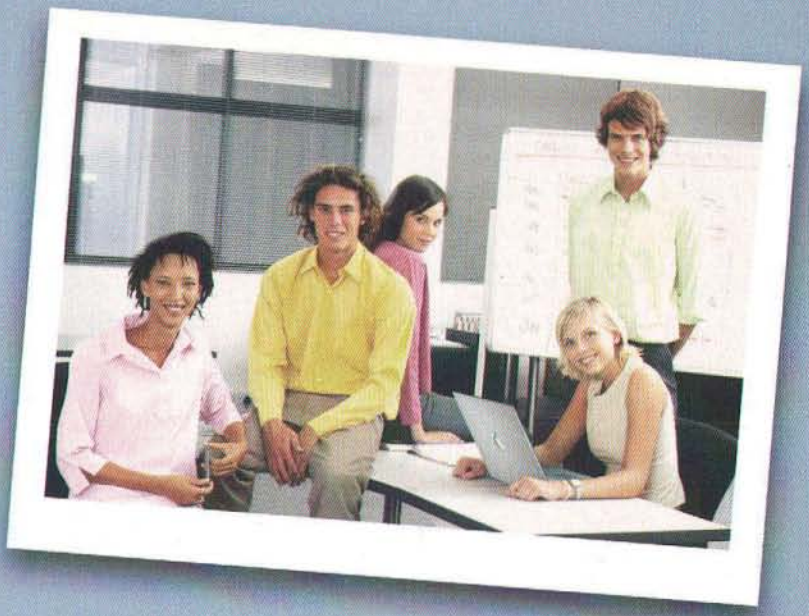
Deciding which tests to write is important, but do not let it paralyze you. First consider which functions should be tested and try to establish the exact requirements of the function. Then you can write some tests that confirm that your function meets these requirements. Since it may be prohibitive to test every possible input, you will need to be judicious about which inputs you use to test your function. The Apple documentation provides some guidelines. As you continue to work on the main application, you will discover cases where the function fails to meet your needs either because the original requirements are not exactly correct, or because the function was improperly coded. Each time this happens, you can add a test. Thinking about how you will test your functions may also affect how you define your functions. A function called solely for the purposes of side effects will be tougher to test than one that produces an output. Similarly, monolithic functions with many tasks will be more difficult to test than smaller functions with a single clear task since you will need to test the monolith with a larger variety of inputs. Finally, the CPlusTest framework does not support the testing of user generated events. I will discuss a (naïve) way to handle this for smaller projects in the section on code coverage below. There are commercial systems for testing user interfaces, but they are beyond the scope of this article.

Target Settings

These instructions come (mostly) from the Apple documentation for Unit Testing with the CPlusTest framework. Start by selecting New Target... from the Project menu. Select Unit Test Bundle from the Carbon section. Choose a name (I chose Unit Tests) and a project (SuperAdd). Voila, a unit testing target. Now you need to make some adjustments to the project configuration so the target will work.

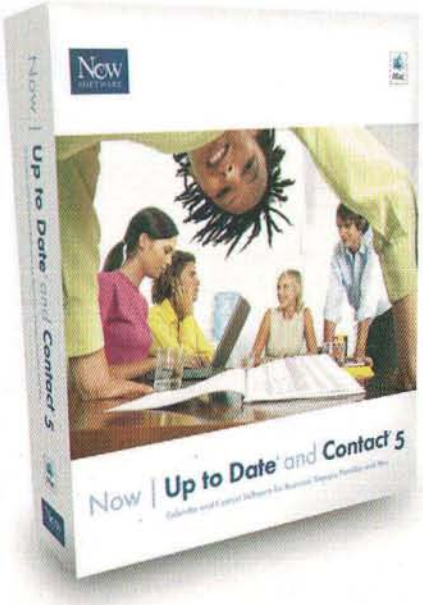
At this point you have to make a decision about whether you want to do unit testing with the Release configuration, the Debug configuration, or a new configuration. The advantage of unit testing the Release configuration is that you will be testing the shipping code. The disadvantage is that you will need to change some of the build settings to use the unit tests and the coverage analysis. These changes may be inappropriate for the shipping product. The disadvantage with testing the Debug configuration is that you are not actually testing your shipping code. You will also not be able to use Zero Link during these

Now **scheduling** and **contact management** for your entire organization.



Now | **Up to Date**® and **Contact**® **5**

Calendar and Contact Software for Business, Groups, Families and You.



Is this project on schedule? When are you available to meet about the systems upgrade? Where are all the field techs today? When was the last time anyone talked to our biggest customer?

Virtually all groups live (or die) by their abilities to meet deadlines and keep track of their customers, prospects, and vendors. Few small companies or even departments of big companies have the tools they need.

Now Up-to-Date & Contact might just be the calendar and contact software for you. It's time-tested and used by more Mac-based companies than any other solution. And it's cross-platform–available for your PC users, too. It's easy to install and manage and simple for your employees to understand and use.

Using Now Up-to-Date & Contact you can schedule meetings for multiple users, view multiple, simultaneous calendars, and reserve rooms and resources. You can share contact information about your customers, prospects and vendors. And using our free server software you can set it up in minutes and share with users in the office or from anywhere with an internet connection.

Call us now at 866-527-0556 or email us at mactech@nowsoftware.com and we'll send you our free evaluation kit, including the book that will make it all easy, "Take Control of Now Up-to-Date & Contact" from Take Control books!

NOW SOFTWARE

Phone: 866-527-0556

Web: www.nowsoftware.com

Copyright © 2006 by Now Software, Inc. All rights reserved. Now Up-to-Date and Now Contact are registered trademarks of Now Software, Inc. Mac and the Mac logo are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. All other trademarks acknowledged.

STOP SHARING!

START FAXING!

Each subscriber receives faxes directly by email as PDF file attachments.

Corporate accounts from 3 to 100+ users available

For more information and a special offer for MacTech readers, visit

www.MaxEmail.com/MacTech

maxemail®

Call: 800-964-2793

builds and this may be important to your development cycle. You could create a new configuration for unit testing (with or without coverage analysis). In larger products, this might be a more appropriate choice. However, for this demonstration, we will go ahead and execute unit tests and coverage analysis with the Debug configuration.

Go to the Targets group and open the information inspector for the Unit Tests target. In the General tab, add a Direct Dependency of the SuperAdd application. This will build the application prior to testing it. In the Build tab, you will need to adjust a number of settings. Make sure that Configuration is set appropriately (in the case of this example, we are setting this up for the Debug Configuration). In the Linking collection, you will need to set the Bundle Loader to your executable. This will allow you to access functions and variables in the original application from your test code. The location for this example is

```
$(BUILT_PRODUCTS_DIR)/$(PROJECT).app/Contents/MacOS/$(PROJECT)
```

In the Unit Testing collection, you need to set the Test Host (the code that your test code will be injected into). In our case, this is the same as the Bundle Loader and so we can use `$(BUNDLE_LOADER)` as the value here. These settings will not affect the SuperAdd application, only the testing code. I have also used the same prefix header for the unit tests as I used for the executable. This prefix header declares a global variable (`gInitialized`) that is used in both sets of code. The SuperAdd code sets this variable to `true` when it is finished with its initialization routine. The Unit Tests code will not start running until this variable has been set to `true`. Using a common prefix header allows both sets of code to see this variable.

If you are building with Xcode 3, you can skip to the next section, entitled Source Code. If you are building using Xcode 2.3, you will need to make some other changes to the targets. In the Unit Tests target, you will want to add the flag `-fno-cxa-atexit` to the Linker's Other Flags in the Linking collection. This is to work around a bug introduced in Xcode 2.3 and 2.4 but fixed in Xcode 3. Now go to the Targets group and open the information inspector for the SuperAdd target. In the Build tab, you will need to adjust two settings. In this case, you are actually setting the build settings for the SuperAdd application. You will probably only want to change these settings in the Debug configuration. In the Linking collection, you need to turn off Zero Link. In the Code Generation collection, you need to turn off Symbols Hidden by Default. I could not find the Symbols Hidden by Default setting mentioned in the Apple documentation. If it is turned on, your Unit Tests bundle will not be able to see the variables and functions you would want to use and you will receive linking errors.

Source Code

Now you need to write the code that runs the tests and the code that implements the tests. Apple supplies a `RunTestsInTimer` class with the CPlusTest framework

documentation that is used to run the tests. I have adjusted the code to create a `CTestRunner` class. When a `CTestRunner` is created, it will create an event loop timer. When the timer fires, the `CTestRunner` checks if the application is initialized. If the application is initialized, it will run the tests, otherwise it will wait until the timer fires again.

RunTests code in CTestRunner.cpp

```
void CTestRunner::RunTests(void)
{
    //gInitialized prevents premature running of tests
    if (gInitialized)
    {
        //prevent a second timer firing while we're doing
the tests
        {
            RemoveEventLoopTimer(myTimerRef);
            myTimerRef = NULL;
            DisposeEventLoopTimerUPP(myTimerUPP);
            myTimerUPP = NULL;
        }

        //run the tests
        {
            TestRun run;
            TestLog log(std::cerr);
            run.addObserver(&log);
            TestSuite& allTests = TestSuite::allTests();
            allTests.run(run);
            std::cerr << "Ran " << run.runCount() << "
tests."
                << run.failureCount() << " failed." <<
std::endl;
        }

        //either quit the application
        //QuitApplicationEventLoop();
        //or show User Interface test instructions
        ShowCoverageWindow();
    }
}
```

The one significant change is that call to `ShowCoverageWindow` instead of `QuitApplicationEventLoop`. Since `ShowCoverageWindow` does not use the CPlusTest Framework's testing macros and classes, but exists solely to obtain complete code coverage, I will discuss it in the section on code coverage below.

I create a testing class for each C module or C++ class used in the main project and use a standardized naming convention: the name of the unit tests associated with the module foobar is called `UTFoobar`. I also organize the unit tests in a source tree underneath the folder `Tests` that mirrors the source tree used for the application. In this case, we have to test the superadd module, so we create a class called `UTSuperadd`. I have also created a module named `UTUI` and it is designed to test the user interface. Like `ShowCoverageWindow` above, it focuses on code coverage and will be discussed later.

The `UTSuperadd` class is used to test the functions defined in superadd. The `UTSuperadd` class is a subclass of `TestCase` (a part of the CPlusTest framework) and contains a number of tests. The class declaration is given below.

UTSuperadd declaration in UTSuperadd.h

```
class UTSuperadd
:public
    TestCase
{
```


Grow Your Revenue
with the
Edgeos Security Services Platform

Your expertise combined with the Edgeos platform enables you to provide Security Services to all of your customers, building loyalty and recurring revenue.

Your security services for Your customers:

- Comprehensive private labeling that enhances your brand and provides a seamless customer experience.
- Simple, scalable pricing that is affordable for your business and ensures you remain profitable.
- Robust scanning engine and extensive reporting to facilitate regulatory/standards compliance.
- Easy to set up, use, grow and manage. You can focus on what you do best.

Visit www.edgeos.com/mactech/ today to view the demo and signup for a free evaluation account.

edgeos
target your customers' security
security services platform
www.edgeos.com/mactech
866.334.3671

GraphicConverter 6

Save 10%

The universal genius
for picture editing

- More than 1.5 million users
- Import of more than 200 graphic formats
- Export of more than 80 graphic formats
- Picture editing
- Document browser
- Slide show and batch processing
- Editing of all meta data (EXIF, IPTC, XMP, ...)
- And much more ...

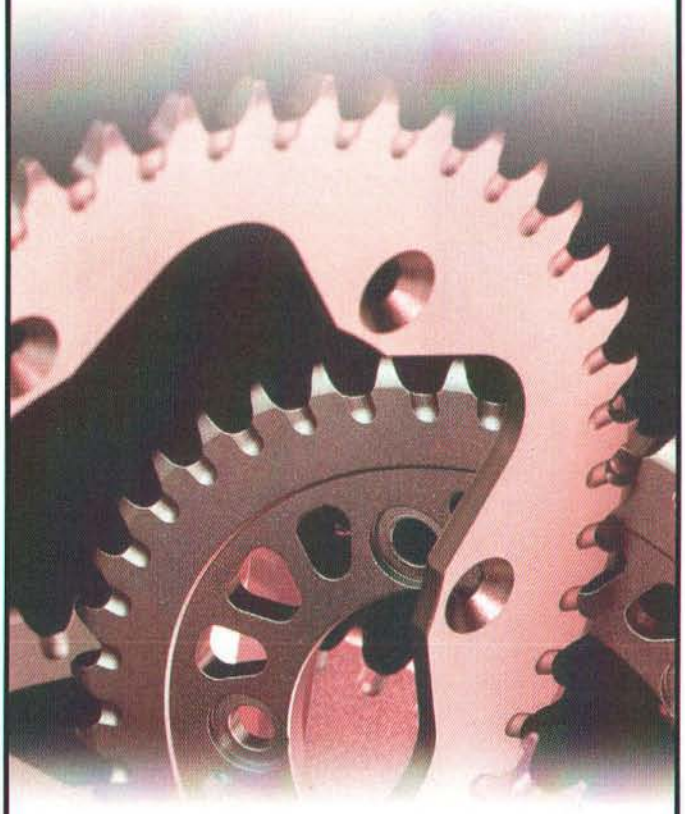
Only $34.95
(Version in the box $44.95)

Save 10% by ordering direct from:
www.lemkesoft.com/mactech

Download the trial version

macforge.net™

MacForge indexes and tracks open source projects that run on the Mac, or are likely to without modification. Thanks to MacForge, there's no need to sift through huge listings of open source that you can't use. With categories, filters, and more, MacForge makes it easy to find what you need.



MacForge:
Your Gateway to Mac Open Source
www.macforge.net

Sponsored by MACTECH

```
public:

    //! This method constructs an UTSuperadd.
    UTSuperadd(TestInvocation* inInvocation);
    //! This method deconstructs an UTSuperadd.
    virtual ~UTSuperadd(void);

    //! This method tests superadd's ability to add two
negatives.
    void TestSuperAddNegNeg(void);
    //! This method tests superadd's ability to add a
negative and a zero.
    void TestSuperAddNegZer(void);
    //
    //  similar tests omitted
    //

};
```

There are two choices when running multiple tests. You could create a single test method that executes all the tests or you can create a number of smaller methods, each of which execute one test. The advantage of the single monolith is that there are fewer tests to register. However, testing will stop at the first failed test. With a number of smaller functions, you will get a log of which tests failed and which tests passed. Since this process is supposed to be automated, I prefer to run a lot of tests in a single batch rather than running until one test fails. It is also often the case that patterns in which tests are failing can lead to hints as to how to debug the code.

The code below demonstrates a simple test to verify that `superadd(-1, -1)` is correct. The definition of the method defines the test, the next line instantiates an object of type `UTSuperadd` and registers the test with the CPlusTest framework. You can use the macro CPTAssert to test assertions. If the input to the macro is false, an error will appear in the build results window.

UTSuperadd::TestSuperAddNegNeg

```
// define the method
void UT_superadd::TestSuperAddNegNeg(void)
{
    CPTAssert(superadd(-1, -1) == -1 + -1);
}

// register the test
UTSuperadd SuperAddNegNeg(
    TEST_INVOCATION(UTSuperadd, TestSuperAddNegNeg));
```

One issue that does not appear in this example is the issue of memory and resource allocation necessary for your tests. It may seem appropriate to make these allocations in a constructor, but that can cause problems since you cannot control exactly when the constructor will be executed (as the objects are static and hence you have no control over when they are created). Instead, allocations should occur in the virtual function `setUp` and deallocations should occur in the virtual function `tearDown`. These functions will be called immediately before and after each test is run. As a result, you know that they will be run after the application has been initialized and before the unit testing has ended.

Cross-platform that really works.

Create your own software. Easily.

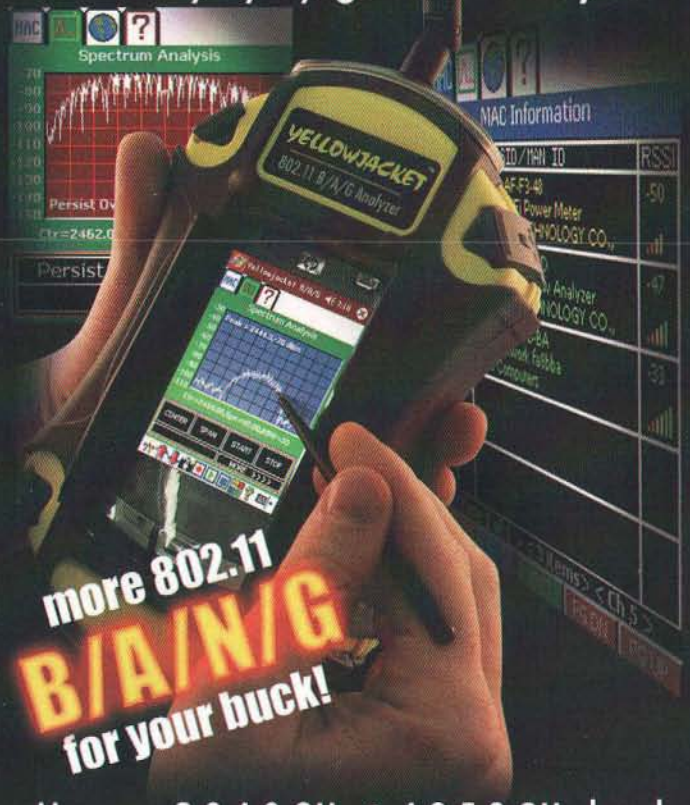
www.realbasic2008.com
Try REALbasic risk-free!
REALbasic includes a 90-day
money back guarantee.

REALbasic is the powerful, easy-to-use tool for creating your own software. At REAL Software, we call it a problem solver. You've probably said, "Wouldn't it be great if there was a little application that ..." REALbasic fills that blank.

REALbasic compiles native applications for Mac OS X, Windows and Linux from one set of source code. Each version of your software looks and works just as it should in each environment. You can even create a Universal Binary with one mouse-click.

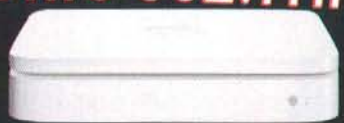
REALbasic is a trademark of REAL Software, Inc.
©2008 REAL Software, Inc.

YELLOWJACKET-BAG
802.11b/a/n/g Wi-Fi Analyzer



- Measure 2.0-4.0 GHz & 4.9-5.9 GHz bands
- Demodulate All 802.11b/a/n/g packets
- User Selectable Power Triggers
- W.I.S.P. Antenna Alignment
- Multipath, MAC, SSID & Absolute Channel
- MAC Security Authorization Lists
- Interference Mapping Software (optional)
- RF Direction Finding (optional antenna)

NOW SUPPORTS 802.11n



Yellowjacket B/A/G is perfect for locating and measuring any device in your 802.11n Hotspot or W.I.S.P. network including Apple's Airport Extreme Base Station.

BERKELEY VARITRONICS SYSTEMS
732-548-3737

Serving the wireless industry for 35 years.

www.bvsystems.com

Running the tests

When you build the Unit Tests target, the application will be built (if necessary) and then the Unit Tests target will be built. As part of the build process of the Unit Tests target, the application will be launched and the tests run. There is no need to choose Build and Run as the tests are run as part of the build process. You can see the Build Results and the Build Transcripts corresponding to running the tests in Figures 1 and 2.

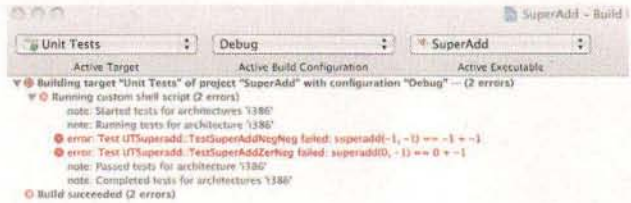


Figure 1: Build Results

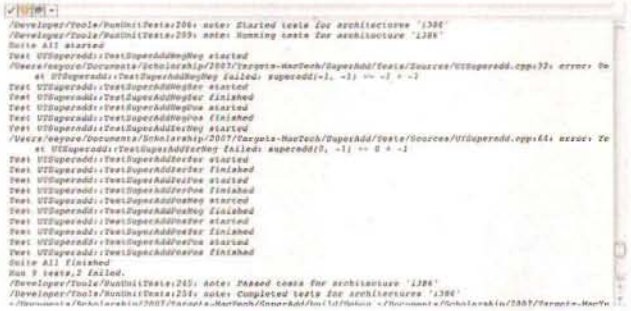


Figure 2: Build Transcript

Failed tests will show up as errors in the Build Results warnings pane. The Build Transcript lists the number of tests run and the number of tests that failed. Assuming your application did not crash, you will also get a note like "Passed tests for architecture 'i386'." This simply means that the application exited normally, it does not reflect whether individual tests were passed. Additional information about which tests ran and whether they passed or failed will also show up in the Build Transcript pane.

One thing you need to be careful about is that the tests will appear to have run even if there was some error in building the application or the unit test bundle. What happens is that an old application or test bundle from a previous build is being run. You should always check the build log to make sure that this did not happen. For important milestone testing, cleaning all targets before running the tests might be a good policy so that you can insure that the tests were run on the most recent build.

Premium Small Business Management and Accounting Software

...and now

Point of Sale for Mac
MYOB Checkout

MYOB

Mind Your Own Business. Smarter.

800 322 MYOB (6962)
www.myob-us.com

MYOB and the MYOB logo are registered trademarks of MYOB Limited.

Coverage Testing

The goal of coverage testing is to execute each command in the source at least once. Like unit testing, a successful coverage test does not mean a bug free program: SuperAdd passes the coverage testing with a phenomenal 100% coverage, but still contains a number of bugs.

The coverage tool provided with gcc is called `gcov`. You can find information about this tool in the GCC documentation (a link is provided in the references). Once you have set up the project to use `gcov` (steps I will present later in this article), you will generate three new types of files. Files with the suffix `gcno` are created when the application is built. They contain the necessary information to link blocks of executable code in the binary with lines in the source files. Files with the suffix `gcda` are created when the application is run. They contain information about which blocks of code were executed. Files with the suffix of `gcov` are created when you run `gcov`. These text files contain an annotated version of your source code where the annotations indicate how often each line of your source was executed. We will not use the `gcov` files directly, but will use the Linux Test Project's coverage tools to create a collection of interlinked html files with the same information. The `lcov` tool (a Perl script) collects the data from `gcov` and creates an `lcov.info` file and the `genhtml` tool uses this file to generate interlinked html files with the coverage information.

One important thing to remember is that `gcov` counts the number of times a line of code was executed. If you are trying to verify that you are executing every instruction, your code layout should contain one instruction per line. Although formatting style is often personal preference or company policy, some formats are more amenable to coverage testing than others. For example, in the first conditional statement below, we cannot tell from the results if `x` was ever incremented, we just know that the equality was tested. The second layout allows us to determine if `x` was incremented.

Conditional statements

```
// here we cannot tell if x++ was executed
if (x == y) x++;
// here we can tell if x++ was executed
if (x == y)
    x++;
```

In addition to possibly adjusting your coding style, trying to obtain 100% coverage may require refactoring your code. If you are finding it difficult to reach some section of code buried inside a larger function, you may decide to write a new function that executes that code. Then you can test this function directly. Whatever you do, don't let the quest for 100% code coverage lead you to poor code writing. The final goal is a well-written program, code coverage is one way to help, but it is not the overall goal.

Getting lcov

The Xcode installer will install `gcov`. You can obtain `lcov` at the website listed in the references. The online documentation for `lcov` is out of date, however the man pages

Mobile email

for Windows Mobile, Palm, Symbian and BlackBerry
powered by Kerio MailServer

Get your email, contacts, calendars and tasks synchronized with your favorite smartphone.
Explore Kerio MailServer, a groupware suite for the office and the road.

Download a 30-day trial version | **www.kerio.com** | **1.408.496.4500** | KERIO

© 2007 Kerio Technologies, Inc. All rights reserved. All other trademarks are property of their respective owners.

appear to be up to date. You will want to place these scripts somewhere convenient. One possibility is in your shell's executable path and another is to package them with the project. In this example, I have created a `Tools` folder as part of the project and added the scripts to this folder (so downloading the project will provide you with the scripts).

The biggest problem with the `lcov` script found online is that it is based on an older version of `gcov`. To reset the coverage testing process, the script attempts to delete all the old coverage data files. The script deletes files with the extension `da`; however, `gcov` now produces files with the extension `gcda`. To fix the `lcov` script, open it in a text editor and then find and replace all occurrences of `.da` with `.gcda`. If you download the `lcov` provided with the project, this has already been done for you.

Target Settings

Again, we need to decide which build configuration we will want to use for coverage testing. If you are testing code coverage while running unit tests, this will be the same configuration you used to build the application that is tested with the unit tests. For this example, we will be adjusting the Debug configuration.

Open the information inspector for the SuperAdd target (not the Unit Tests target). In the Code Generation collection, turn on Instrument Program Flow and Generate Test Coverage (these options will create the `gcno` and `gcda` files). In the Linking collection, add `-lgcov` to the Other Linker Flags (this option will link in the `gcov` library). Notice that you do not need to adjust any settings for the Unit Tests target. You are not testing coverage of the code in the unit tests.

Shell Script

The unit tests are run in a Run Script phase of the Unit Tests target. Go to the Targets pane and disclose the phases for the Unit Tests target. Replace the Run Script phase script with the following code.

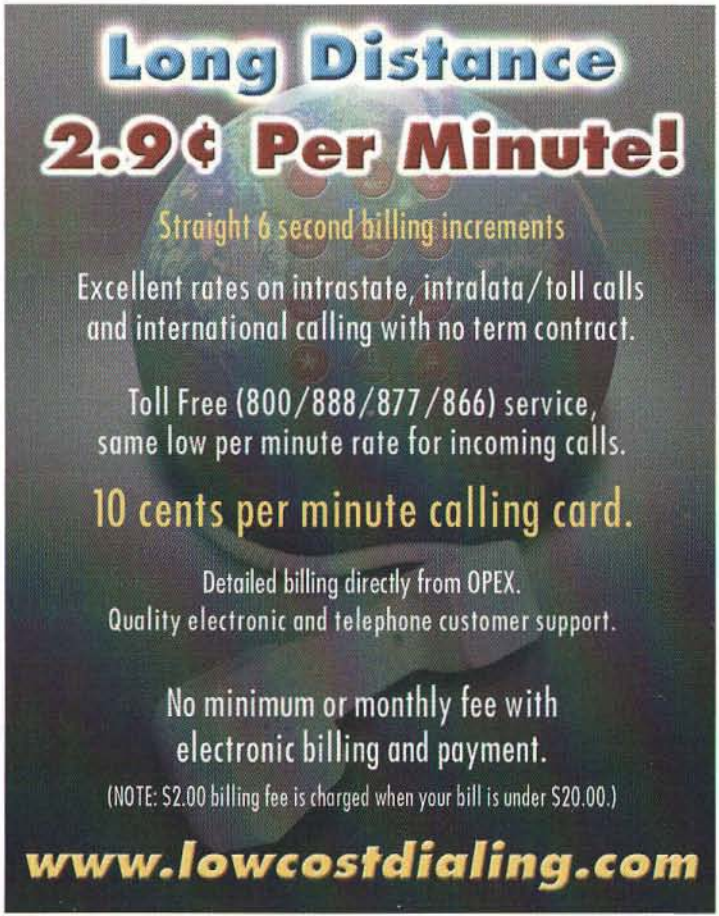
Run Script Phase for "Unit Tests" Info

```
source ${PROJECT_DIR}/Tools/lcov.sh
```

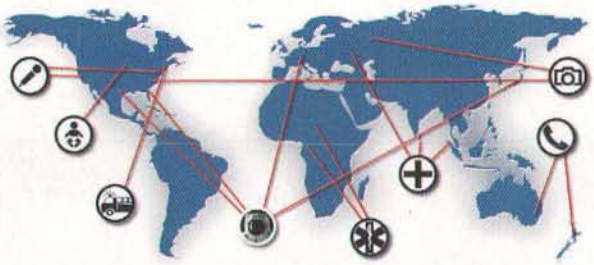
The shell script that is actually executed is shown below. I have used a prefix of MONSTERWORKS in the shell script to prevent clashing shell environment variables. In the script listed below, I abbreviated this to MW. Unfortunately, even with this abbreviation, the script contains some very long lines. The version below tries to break the lines. The character ¬ along with any following white space should be removed (or read the script included with the project).

lcov.sh

```
# the name of the application
MW_APP_NAME=SuperAdd

# the target that builds the executable
MW_TARGET_NAME=${MW_APP_NAME}

# the configuration in which we do unit testing/coverage
analysis
```



SyncDeK



Any Time,
Any Where Data

- **Mobility**: Data delivered any time, anywhere in the world.
- **Synchronize data** between FileMaker databases and now SQL data sources, easily and securely, no matter where they are.
- For offline laptops or online home offices and multiple office locations – each local copy of the database synchronizes its data with the others, **automatically or on demand**.
- SyncDeK **encrypts** all data transmissions.
- SyncDeK works with your databases behind your firewall to **prevent intrusion**, or keeps all database copies completely decentralized.
- Protect mission critical data with SyncDeK's continuous offsite differential **data backup**, warm standby and fail-over servers.
- **Sync servers** with client-free server replication.
- **Version Update Manager** allows you to update all copies of the database without requiring installation or any action on their part.
- **Call for demo, trial and best pricing options.**

WorldSync

877.548.4920 toll-free
510.548.4920 international
www.worldsync.com

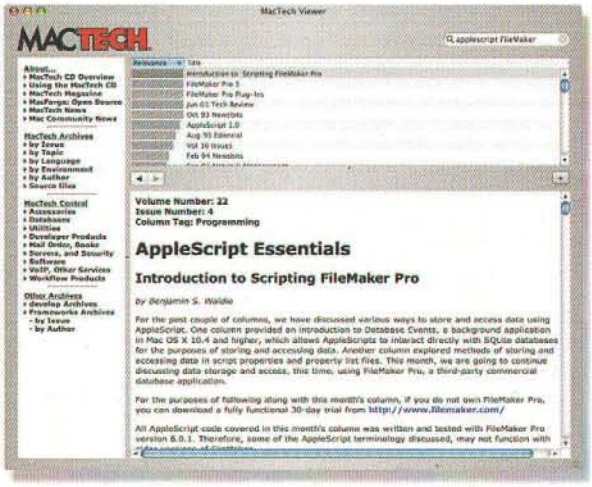
```
MW_CONFIGURATION=Debug

# path to the lcov tools
MW_LCOV_PATH=${PROJECT_DIR}/Tools

# where the object files for the application will be found
MW_OBJ_DIR=${OBJROOT}/¬
    ${MW_APP_NAME}.build/${CONFIGURATION}/¬
    ${MW_TARGET_NAME}.build/Objects-normal/${NATIVE_ARCH}

# we only execute the coverage test if we are using the
'Debug' configuration
if [ "${CONFIGURATION}" = "${MW_CONFIGURATION}" ]; then

    # clean out the old data
    ${MW_LCOV_PATH}/lcov ¬
        -directory ${MW_OBJ_DIR} -zerocounters

    #remove the old report
    pushd ${OBJROOT}/${CONFIGURATION}
        if [ -e lcov ]; then
            rm -r lcov/*
        fi
    popd

    # run the unit tests
    "${SYSTEM_DEVELOPER_DIR}/Tools/RunUnitTests"

    pushd ${OBJROOT}/${CONFIGURATION}

        # create the coverage directory
        if [ ! -e lcov ]; then
            mkdir lcov
        fi

        #analyze the coverage data
        ${MW_LCOV_PATH}/lcov ¬
            -directory ${MW_OBJ_DIR} ¬
            -capture -output-file lcov/lcov.info

        # create the html pages
        ${MW_LCOV_PATH}/genhtml ¬
            -output-directory lcov lcov/lcov.info
        # open the coverage analysis
        open lcov/index.html

    popd

    # clean up
    ${MW_LCOV_PATH}/lcov ¬
        -directory ${MW_OBJ_DIR} -zerocounters

fi
```

Although it appears long and complicated, the steps are fairly simple. If we aren't using the correct configuration, we simply skip the script. Otherwise, we start by removing any of the coverage results from the previous run of the script. Be careful with the recursive `rm` command and confirm that you really are removing the files from the correct directory. After this, we run the unit tests. Next we run `lcov` to generate the coverage results and `genhtml` to produce the HTML pages. We finish by opening up the HTML pages and cleaning up after ourselves.

Now when you build the Unit Tests target in the Debug configuration, the application will be built (if necessary), the application will launch, and the unit tests will run and the application will quit. Then `lcov` and `genhtml` are executed and the results of this are opened so that you see a window like that shown in Figure 3.

The MacTech CD - Volumes 1.01-23.06 is packed with more than ever before -- over 2900 articles from more than 260 issues (**1984 - June 2007**) written by over 840 experts, all 29 issues of Apple's develop, 21 issues of FrameWorks magazine, 100+ MB of source code, MacTech Viewer, working applications, full documentation, demos for techs, ***and more!***

Everything is displayed in the all new, very fast, very searchable ***MacTech Viewer!*** A new application that has been designed specifically with Techs in mind. Search quickly through *20 plus years* of great information provided by MacTech. Information to make your life easier.



Requires Mac OS X v. 10.4.5 or later

Toll Free 877-MACTECH, Outside US/Canada: 805-494-9797 • http://www.mactech.com/cd/

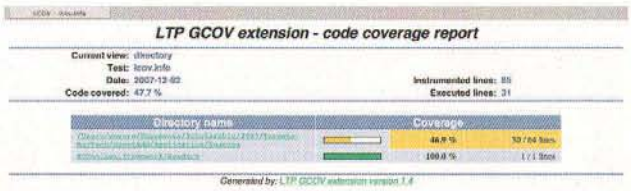


Figure 3: Coverage Overview

There is an inline function in `Headers` that causes it to show up in the coverage analysis, but we are primarily interested in the `Sources` folder. Following that link and then the link to the results from `main.cpp` leads to a page shown in Figure 4.

Figure 4: Missed Lines

Blue lines were executed and orange lines were not. If the line is uncolored, then it does not contain executable code (commands that span multiple lines have the last line highlighted). In this case, it is the window event handler that is not being called. This isn't surprising since we never interact with any windows in the program.

Testing the User Interface

Automated testing of the User Interface is beyond the ability of the CPlusTest framework. However, we can interject some supervised user interface testing with the project. The code in UTUI works for simple user interfaces. It opens a utility window that leads the user through the steps they should take to exercise the code. One step for SuperAdd is closing the window and the utility window for this step is shown in Figure 5.

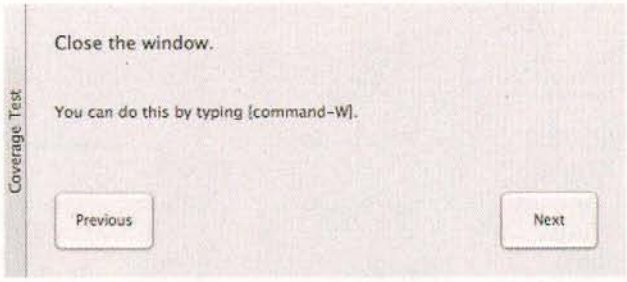


Figure 5: User Interface Testing

To achieve 100% code coverage of SuperAdd, you should comment/uncomment the lines in `CTestRunner::RunTests` to invoke `ShowCoverageWindow` instead of `QuitApplicationEventLoop`, build the Unit Tests target, switch to the application, and follow all of the instructions in the utility window.

Conclusion and References

I think I've run out of space, but hopefully you will be able to implement some of these ideas in projects of your own. The following is a list of references that have been mentioned in the article.

Documentation:

You can find a MacTech article about how to set up a target to use doxygen to document your code at:
http://www.mactech.com/articles/mactech/Vol.20/20.03/Documentingyourcode/index.html

Unit Testing:

You can find information about unit testing with Xcode at:
http://developer.apple.com/documentation/DeveloperTools/Conceptual/UnitTesting/UnitTesting.html

There is also a tutorial on using unit testing and coverage analysis at Chris Liscio's Boo-urns Log at:
http://www.supermegaultragroovy.com/Software%20Development/xcode_code_coverage_howto

GNU Documentation:

You can find documentation about gcov along with the rest of the gcc tools at:
http://gcc.gnu.org/onlinedocs/

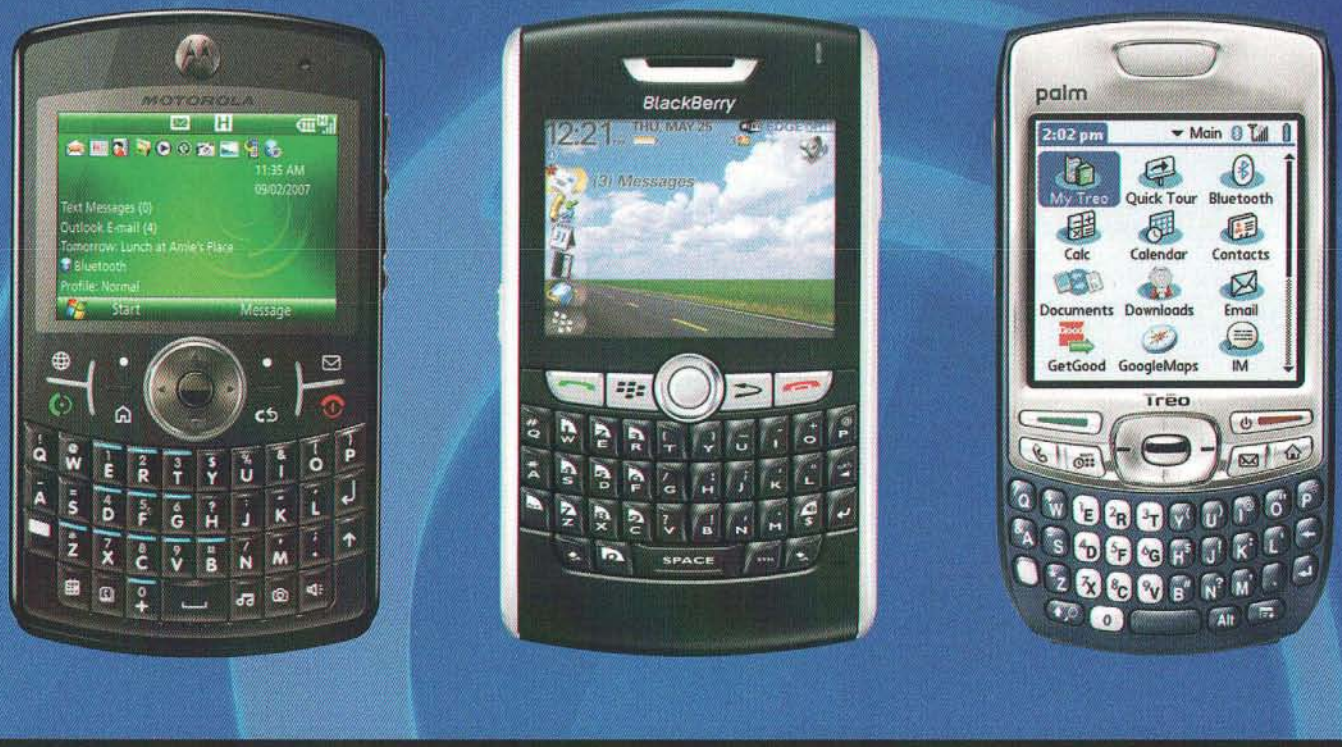
Linux Coverage Tool:

The Linux coverage tools can be found at:
http://ltp.sourceforge.net/coverage/lcov.php

About The Author

Aaron Montgomery teaches mathematics at Central Washington University. He also enjoys hiking, mountain biking, and alpine skiing. You can reach him at eeyore@monsterworks.com.

The Missing Sync™



Reliable Handheld Synchronization for Mac

Thanks to **The Missing Sync**, Mac users are no longer second-class citizens when it comes to keeping their mobile devices in sync. Whether it's a **Windows Mobile 6** device, a **BlackBerry** or even a **Palm** Treo, there's a Missing Sync product made to connect and synchronize that device with Mac OS X.

Syncs Contacts, Calendars, Tasks

- Supports Address Book, iCal, Microsoft Entourage 2004
- Outstanding field support, even syncs Address Book contact photos
- Supports calendar event reminders and detached events
- Sync Services-savvy for syncing with third-party apps

Complete Notes Syncronization

- Includes Mark/Space Notebook for Mac to create, edit, categorize, sort and search through notes
- Also supports Microsoft Entourage 2004 and Bare Bones Yojimbo
- Full support for Sync Services and .Mac syncing between Macs

iPhoto and iTunes Integration

- Imports photos and videos from mobile devices into iPhoto albums or folders in the Finder
- Resizes and downloads selected iPhoto albums to the device for handheld viewing and slideshows
- Downloads DRM-free music and podcasts for mobile playback

The Missing Sync for BlackBerry, Palm OS or Windows Mobile is available in single-user licenses for $39.95 or in multi-user packs for any size organization, The Missing Sync family of products provide a Mac-centric synchronization solution second to none.

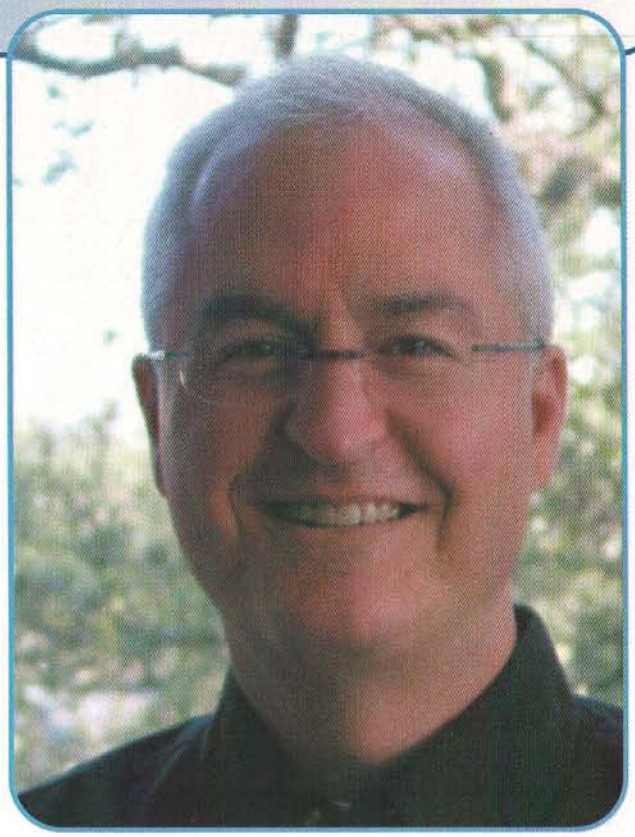
Visit www.markspace.com today to see how easy it is to sync the latest smartphones and mobile devices with the Mac.

mark|space

Geoff Perlman on REALbasic

Catching up with Geoff and REALbasic 2008

by Norman Palardy



INTRODUCTION

At MacWorld 2008 REAL Software announced the first release of REALbasic for 2008. MacTech Executive Editor, Edward Marczak and columnist Norman Palardy had an opportunity to ask REAL Software CEO Geoff Perlman about the announcement, REALbasic and REAL Software.

THE INTERVIEW

Norman Palardy: The press release went out: It was announced on the various boards, MacNN and MacWorld, it's on your website, it's been published in your forums and sent to your mailing list. But, it's a press release so it only says so much. Can you tell us what's new in REALbasic 2008 R1?

Geoff Perlman: The big thing is introspection. It is the most heavily requested feature of all time. That is what's appearing in this release. There is always a lot of maintenance. There are a few minor features but introspection is the big thing.

NP: For somebody who might not be familiar, what is the current pricing of REALbasic and how do you handle updates?

GP: The standard edition is $100 and the professional edition is $500. The difference is that the professional edition, among other things, adds access to database servers and cross platform compilation. Actually, I should back up because we just made a change with this release. We are no longer calling it the standard edition. It is now called the Personal edition, and this is to help distinguish between editions. People ask us "why don't I use one or the other"? If you are writing software for yourself, then chances are the personal edition is appropriate. If you are writing software for other people, the professional edition is probably what you need.

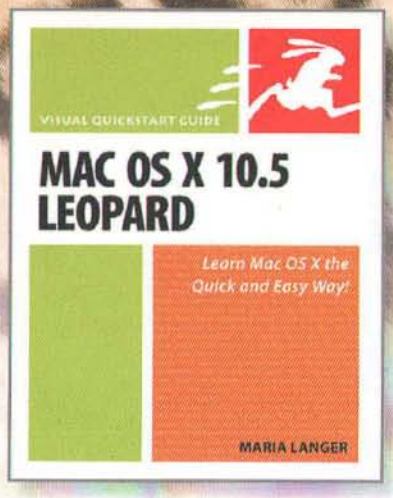
Obviously there are times when you might be writing software for other people and the Personal edition would work. But most of the time we find that when people are writing software for somebody else they need some feature that is in the Professional Edition.

NP: That is an interesting name change. I wasn't aware of that, but it should help give people better clarity, as you said, as to what the intended audience is.

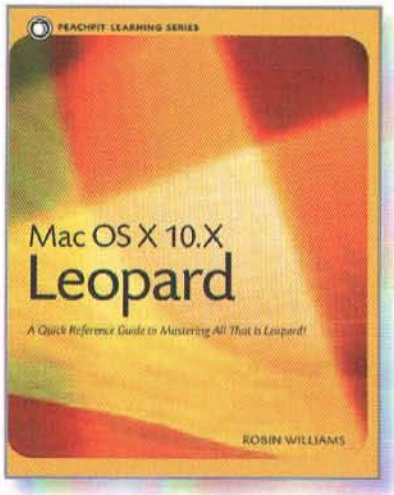
NP: Some time ago you guys switched from the traditional dot-1 dot-2 dot-3 kind of release naming to a different schedule and a different naming scheme. How frequent are updates nowadays?

GP: We release every 90 days. Occasionally it is less than 90 days. We made that change for a number of reasons. First of all, our customers are developers. Even if they are one hour a week developers, versus 40 hour week, they are developers and we need to be able to respond more quickly to what they want. We felt having a year, or a year and half or two year development cycle doesn't make sense. We can't respond to the market quickly enough that way. The other thing is that the smaller you make a project in scope or in time, the more likely you are to succeed. So, for us internally, by breaking our delivery schedule down into ninety day segments, we are much more successful at meeting our deadlines and getting things done on time than we were before. So it worked out really well.

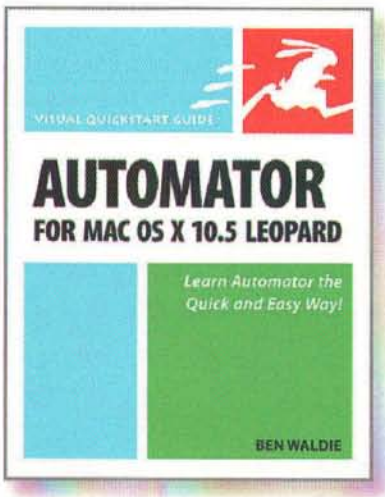
NP: At one of the REAL World conferences you had mentioned that REALbasic has surpassed 100,000 users. That was over a year ago. What is the user base up to now? You have to be well beyond that.



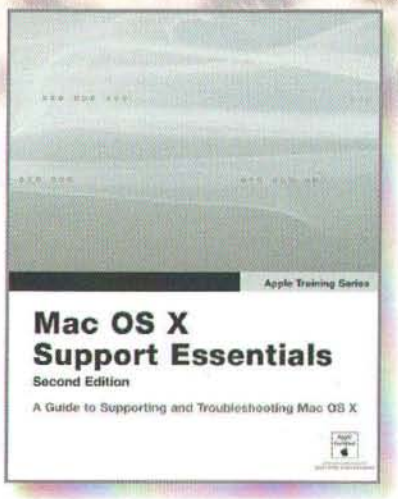
Mac OS X 10.5 Leopard: Visual QuickStart Guide
ISBN: 0321496000
Maria Langer
Available now!



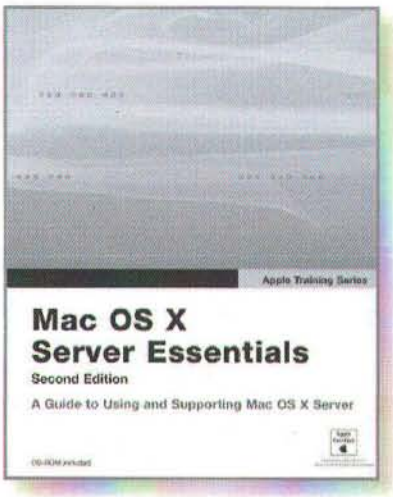
Mac OS X 10.5 Leopard: Peachpit Learning Series
ISBN: 0321502639
Robin Williams
Available now!



Automator for Mac OS X 10.5 Leopard: Visual QuickStart Guide
ISBN: 0321539354
Ben Waldie
December 2007



Apple Training Series: Mac OS X Support Essentials, Second Edition
ISBN: 0321489810
December 2007



Apple Training Series: Mac OS X Server Essentials, Second Edition
ISBN: 0321496604
December 2007

Are you ready for Leopard?

Learn the ins and outs of all the new features in Mac OS X 10.5 Leopard, including Time Machine, Spaces, Quick Look, Automator, and more. Whether you're an experienced Macintosh user ready to hone your trouble-shooting skills, or you're leading a group of beginners who need help getting started, pick up a Peachpit book today to get the most out of the powerful features in Leopard.

BUY NOW AND SAVE!

Purchase or preorder your copy today and **save 35%** off the list price, plus free domestic shipping. Simply go to **www.peachpit.com/leopard** and enter coupon code **PP-LEOPARD** when you reach the checkout page.

Peachpit

Apple Certified

GP: At this point we have over 125,000 users.

GP: I will tell you that we track the percentage of new customers we get each month versus renewals from existing customers. And the percentage of new customers has been climbing almost every month, which is really great.

NP: And are they renewals? Obviously those numbers tell you about broader appeal, lots more people trying it out and what not. But the update numbers must tell you something too.

GP: Yeah they do. I think it works better on the new model. With the old model, people would look at each individual release rather than thinking 'I just want to make sure I'm up to date'. The new model actually allows us to do a lot more maintenance on the existing code. It's a change of philosophy. Rather than feeling like they are buying this particular update or this particular new version, they are buying the next six months or the next twelve months of updates or versions. As a result of that change in their thinking about what they are buying we can get away with pouring more time into maintenance. Before there had to be lots of new features in each release because if there wasn't, there wouldn't be a reason for people to upgrade.

I am a big believer that the road to happiness is managing people's expectations. If you want people to be happy, manage their expectations. Get their expectations to an appropriate level and, of course, you have to meet those expectations. And I think that's where you can get into trouble by having a big discrepancy between what people are expecting and what you are delivering. So one of the things we keep trying to do is bring those two things together.

NP: And you feel the new release model is helping you do that better?

GP: Oh absolutely. I would never go back. It works so much better. Look at it this way: Suppose you knew we were working on a feature and we got close to the release date and decided it's just not ready so we are not going to put that in this version. We ship it and you know that the next version is a year a year and a half away. And you were waiting for that feature. With our rapid release model, worst case it's 90 days away. That's not very far. So really it allows us to produce a better, higher quality product than when we were using the traditional model.

NP: I think there are a lot of things that have really come along nicely with the new model and, like anything else, there have probably been growing pains to get to that point.

Ed Marczak: Imagine if Microsoft did this with Office. You just pay this one fee and then there were continual releases. I think that would be a much more successful model for them.

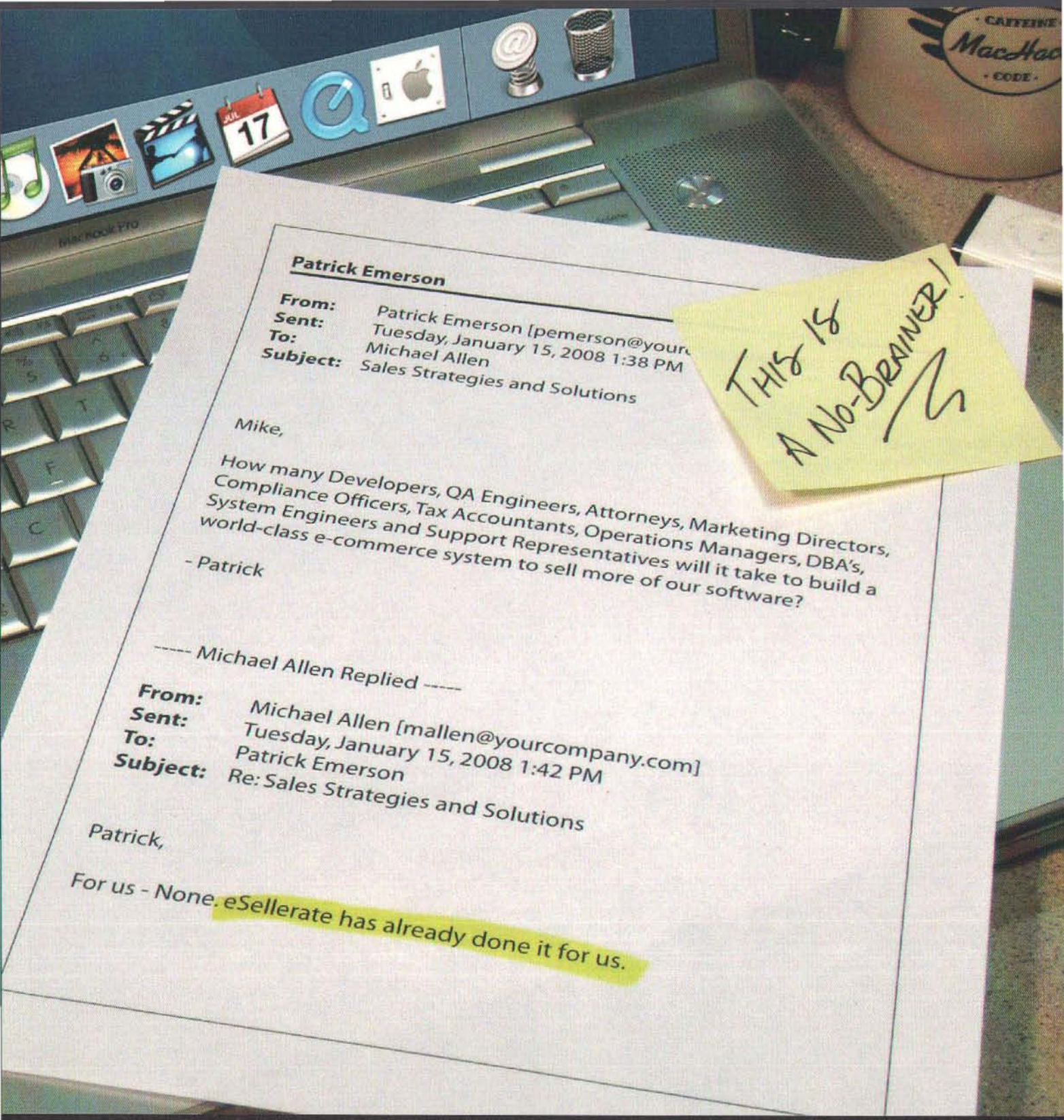
GP: Well, in fact, Microsoft has announced they are moving away from the monolithic release. They are actually moving to this model.

EM: Oh really? Wow!

GP: Probably not every 90 days, though.

NP: Let's be realistic, software is never perfect. It is never bug free so it is constantly evolving and gaining new features and gaining new fixes. This model fits better with that reality.

GP: Everyone would love to have code that is bug free but they don't want a static product either. Static, bug free code is extraordinarily expensive. For example NASA claims that the code



eSellerate. Complete.

Begin defining your strategies at http://www.esellerate.net/mactech

eSellerate, the same team that brought you Installer Vise.

eSellerate is a registered trademark of MindVision, Inc. - a Digital River Company.

Mac Universal

that runs the space shuttle is bug free. But they also say that it costs $25,000 per line of code.

NP: That's a lot of money to be spending on one line of code.

GP: Right.

NP: Over the course of the past year the company has had some changes. You have had personnel changes, and now a product name change. How are those things impacting your ability to deliver as a company? Or are they having an impact. I mean loosing developers has got to have in impact.

> *"I am a big believer that the road to happiness is managing people's expectations. If you want people to be happy, manage their expectations."*

GP: We have only lost one developer in the last year and he actually still continues to do work for us on contract. Honestly, like the product, our development team evolves. And that's not a bad thing because what we need in our development team changes and developers don't always keep up with changes in technology. [Take] for example when we added support for Mach-O, which is the one of the two executable formats, and really the only one now that Apple supports. We started supporting that years before that change was made and there was a lot of debate in the engineering team because they really preferred the old PEF format and they believed it was better.

I said 'guys,it doesn't matter if it's clear or not' because Apple is saying that Mach-O is the blessed format so it's a pointless debate whether Mach-O or PEF are the better format. Now we don't support PEF because with Mac on Intel, Mach-O is the only format. Sometimes people's attitudes about technology don't change with the tide and if that is the case, if they leave and we bring in new developers that have a different take on things, we're going to make sure that the people coming in are looking at development the way that we do. Honestly, I think one of our strengths with REALbasic has been that we have been successfully able to keep up with technology. If you bought version 1 of REALbasic it ran only on a Mac and it was PowerPC and 68K. Now it runs on Mac OS X on Intel. It runs on Linux and Windows up to Vista. You can build console apps. We have abstracted our customers from lots and lots of those platform details.

NP: And it was only recently that you quit supporting OS 9. Realistically you have to at some point. You supported it long after Apple said 'OS9 is dead'

GP: The other thing is that it's important to recognize that if you try to develop an application for 100% of your target market you won't make anybody happy. The application will be too feature rich. There will be too many options and it will be too complex. So our attitude, and I think this is Apple's attitude as well, is make 80% of your potential customers happy you will have a much better product than going for 100%.

NP: Do you find when you are building a cross platform tool like you are that you have to make platform specific compromises in any way?

GP: No, we try to make it so that when we look at any piece of technology we say, "how do we make this cross platform?" Generally speaking we don't have to make those compromises if it's supported on more than one platform. Going back to the 80/20 rule, 80% of the functionality that 80% of your customers want is probably going to be supported on all three platforms. So we just have to make sure that we provide an API that makes sense across all three. But we're not afraid to add something that is platform specific either and let the developer make the decision as to whether to use that or not. AppleScript on the Mac or the Registry on Windows are other examples. Generally speaking we don't have to make those kind of compromises because it's either cross platform or it's platform specific. If it's cross platform we can usually come up with a good API and if it's platform specific then it's not a problem.

"I said 'guys,it doesn't matter if it's clear or not' because Apple is saying that Mach-O is the blessed format."

NP: I was just wondering about the trade-offs. When you look at some of the other tool kits they always seem to trade something off. Or they don't use native controls. I was just wondering if you encountered that in developing REALbasic as a cross platform framework

GP: I think that if you recognize that you should design for 80% of your potential customers then you give up very little, if anything. I think it is when you try to make everybody happy and pitch your solution as the be all end all solution, the Holy Grail solution, that's when you run into trouble.

NP: Over the years that we've known each other and dealt with each other, I might have asked you 'who do you see as your primary customer, is it a Mac user, is it a Windows user is it a Linux user' and you've always said 'That's not necessarily the way we perceive ourselves. We're a cross platform company so they are all our users". Right?

GP: Our user base is pretty broad but it's basically people that want to build cross platform applications and they want to do it quickly. They don't want a big learning curve. They want to be abstracted from all the platform details and I think we do cross platform better than anybody in the world. Frankly, I'm not afraid to say that. I ask people, "think of a cross platform tool set that does a better job than we do". I don't think there is one out there.

NP: There aren't a lot of them to start with. Which ones have a rapid application development environment like REALbasic? The list gets pretty short really really quickly and that is one of the attractions of REALbasic. I'm curious about the product name change. Do you see that as targeting a particular kind of developer? I think people who develop software for a living have this notion of "professional developers" versus "hobbyists".

GP: I think what it is, is that "standard" as a name was a mistake from the very beginning. If someone says 'This is a standard version of anything' that implies this is the version you should buy. It's the standard. And that's really never what we meant and we

really should have taken a different approach in the first place. Over the years what we've found is our users tend to be people writing software for themselves or people writing software for somebody else. We recognize the people that are writing for themselves buy standard and the people writing for others buy pro. The people writing software for other people are generally being paid to write software therefore they are professionals. So the professional version makes sense.

The problem is that, honestly, it is a marketing thing. It's like some corporate IT guy has to explain why he needs the $500 version when there is $100 version available. "Standard" sounds like that is what you need; it's the standard version. If it's personal versus professional he can go to his boss and say this one is clearly for individuals, and that the publisher (us) is telling you to buy the professional if you are building for other people. The IT guy builds for other people therefore he needs the professional edition. That's a very easy way for people to pick the right product for them.

But honestly if the question is 'do hobbyists programmers exist I can tell you they definitely do.

NP: I have no doubt they do but I think a lot of people who use that moniker use it in a derogatory sense and that's an unfortunate thing because there are certainly a lot of people who program as a hobby and that doesn't make them ineffective or unskilled.

GP: No, no. What I have found is that hobbyists are people who describe themselves by saying "I don't do this professionally. No one is paying me to do it. I just do it on my own". Believe it or not, we get a lot of psychologists that buy REALbasic and they're developing the software pretty much for themselves although it is going to be used by other people. I guess that is developing for other people, but they will build software to do psychological testing and often times that's going to be on one platform. It doesn't need to be cross platform, so they're sort of the exception, where they are building software for other people but really they only need one platform, they don't need SSL or database access or that kind of stuff.

NP: REALbasic has been around for...

GP: Ten years this July, 4th

NP: Over the course of ten years, one of the things that hasn't really sort of sprouted up of its own accord is a big third party market. I'm not sure how it came about with a thing like Visual Basic. Do you see that as being important to the overall success of REALbasic as well having a big vibrant successful third party market?

GP: Well, I will say this, the thing I think is important to the success of REALbasic is that customers need to be able to get all their needs met. So it's not a question of a third party market or not. Having all their needs met is what makes the product successful. Or is at least one of the elements that makes the product successful. I think in the past we have taken on way too much and tried to put everything and the kitchen sink into the product and that has limited the opportunities for third party developers. But if you've been watching the release notes for the last couple of releases you've been seeing the word deprecated showing up, and what you are going to see in the future is that we're going to start trimming down the product a little bit. The features that only a small group of users need are going to become more third party opportunities so we can focus on the core product and make it even better. And, honestly, I think that is a mistake we made in the beginning was not recognizing that we really should stick to the core product and try to develop the third party market. Now, there is a fine line there.

When I worked at 4D long ago there was a big third party product called AreaList. It was a grid control and they never built a good grid control into 4D. They were afraid they would upset the makers of AreaList because anyone who was doing any serious development with 4D used AreaList. They were basically going to screw the developer of AreaList if they built something in that did the same thing. I think that was a mistake because a lot of customers looked at 4D and didn't know much about third party market. So what happened was people said "Well it's kinda weak when it comes to grid control". So you have to pick your battles. With things that are really important, it's the 80/20 rule again. If it's important to 80% of your customers, it probably means it needs to be built into the product. If it's less than 20%, probably that is a good third party opportunity. And we're going to really try in the coming years build up the third party market and partner with third party developers so that we can get them the exposure that they need to our customer base to help them be successful. That's a win win situation.

NP: Actually you just led right into my next question which is exactly that: If it's important, is there something you see REAL needing to do to help bring that to fruition ?

GP: Well, you've already seen that we are selling books on our website, Soon we will be selling RB Developer Magazine. We're probably at some point going to have a third party section of the website. When people are evaluating REALbasic we want to make sure that they know there is this set of third party tools out there.

EM: To know there's a vibrant community.

GP: Exactly: knowing that the community is there, knowing the third party community is there helps us and it also helps those communities to grow and to continue to be vibrant. So it's totally a win win situation. I can't tell you when that is going to happen but it is definitely something that's very important to us going forward.

NP: Thanks for taking the time and letting us do this.

GP: No problem. Thank you.

About The Author

Norman Palardy has worked with SQL databases since 1992, and has programmed in C, C++, Java, REALbasic and other languages on a wide variety of platforms. In his 15+ years of IT experience, Norman has developed innovative and award-winning applications for TransCanada Pipelines, Minerva Technologies (now XWave), Zymeta Corporation, and the dining and entertainment industry. He holds a BSc from the University of Calgary in Alberta.

Advertiser/Product Index

Addlogix (formerly CompuCable Mfg. Group) 44
Akor 12
Aladdin Knowledge Systems, Inc. 53
Allume Systems, Inc. 1
Ambrosia Software Inc. 15
Aqua Connect, Inc. 27
Berkeley Varitronics Systems, Inc. 64
Brad Sniderman 38
Cynergy Data 22
EazyDraw (Dekorra Optics, LLC) 24
Edgeos, Inc. 61
eSellerate/MindVision 75
Faronics Corporation 26
Fontlab Ltd. 21
Freeridecoding 46
Future Media Concepts 43
Hewlett-Packard Company ..... IFC
IGC, Inc. / MaxEMail.com 60
IGG Software, LLC 4
iSkin Inc. 18
Kerio Technologies Inc. 66
LANrev LP 37
LassoSoft LLC 56
Lemke Software GmbH 61
LithiumCorp 51
Mac Rentals, Inc. 23
MacForge.net 74
MacMall 2-3
MacResource Computers & Service 41
MacSpeech, Inc. 28
MacTank 52
MacTech Magazine 69
Mark/Space Inc. 71
MARWARE 25
MDG Computer Services, Inc. 39
Microsoft ..... BC
Mushkin 17
MYOB US, Inc. 65
Now Software 59
OlympicControls Corp. 33
Other World Computing 9
Other World Computing 55
Ovolab 40
Parallels Inc. 31
Peachpit Press 73
Powerbookmedic.com 45
ProjectWizards 32
RAMJET 35
REAL Software, Inc. 63
Sans Digital 76
Seapine Software, Inc. 13
Small Dog Electronics ..... IBC
Stellar Information Systems Ltd. 42
SupaCam / Nisis 19
TechRestore 50
Toon Boom Animation 29
Trango Broadband Wireless 47
Utilities4Less.com 67
Walter Karl, An InfoUSA Company 74
WIBU-SYSTEMS AG 11
WorldSync, Inc. 68
Yazsoft.com 77
ZAGG Inc (dba ShieldZone) 67

AccountEdge • MYOB US, Inc. 65
Aqua Connect Terminal Server • Aqua Connect, Inc. 27
BookEndz • OlympicControls Corp. 33
CodeMeter • WIBU-SYSTEMS AG 11
DAS, NAS, and SAN Storage Solutions • Sans Digital 76
Deep Freeze • Faronics Corporation 26
EazyDraw • EazyDraw (Dekorra Optics, LLC) 24
eMail Appending • Walter Karl, An InfoUSA Company 74
eSellerate • eSellerate/MindVision 75
eTailer • MDG Computer Services, Inc. 39
Font Editor • Fontlab Ltd. 21
Gateway • Cynergy Data 22
Graphic Converter • Lemke Software GmbH 61
HASP • Aladdin Knowledge Systems, Inc. 53
iBank/iBiz • IGG Software, LLC 4
invisibleSHIELD by ZAGG • ZAGG Inc (dba ShieldZone) 67
iSkin • iSkin Inc. 18
IT Training • Future Media Concepts 43
Kerio Server Software • Kerio Technologies Inc. 66
KVM Switch • Addlogix (formerly CompuCable Mfg. Group) 44
LANrev • LANrev LP 37
Lasso • LassoSoft LLC 56
Law Offices • Brad Sniderman 38
LITHIUM NMP • LithiumCorp 51
Long Distance Phone Service • Utilities4Less.com 67
MacMall • MacMall 2-3
MacResource Computers • MacResource Computers & Service 41
MacSpeech Dictate • MacSpeech, Inc. 28
MacTech CD • MacTech Magazine 69
MailTank • MacTank 52
maxemail.com • IGC, Inc. / MaxEMail.com 60
Memory • Mushkin 17
Memory • RAMJET 35
Memory Products • Other World Computing 9
Mercury Storage Solutions • Other World Computing 55
Merlin • ProjectWizards 32
MiniMax Keyboard • Akor 12
Missing Sync • Mark/Space Inc. 71
Now Up-to-Date • Now Software 59
Office 2008 for Mac • Microsoft ..... BC
Open Source Directory • MacForge.net 74
Parallels Desktop for Mac • Parallels Inc. 31
Peachpit Press • Peachpit Press 73
Phlink • Ovolab 40
Powerbookmedic.com • Powerbookmedic.com 45
Printers • Hewlett-Packard Company ..... IFC
Project X • MARWARE 25
REALbasic • REAL Software, Inc. 63
Recovery Utilities • Stellar Information Systems Ltd. 42
Rentals, Computer • Mac Rentals, Inc. 23
Repairs and Upgrades • TechRestore 50
Security Services • Edgeos, Inc. 61
SmallDog.com • Small Dog Electronics ..... IBC
SmartBackup • Freeridecoding 46
Spectrum Analyzer • Berkeley Varitronics Systems, Inc. 64
Speed Download • Yazsoft.com 77
StuffIt • Allume Systems, Inc. 1
SupaCam • SupaCam / Nisis 19
SyncDek • WorldSync, Inc. 68
TestTrack • Seapine Software, Inc. 13
Toon Boom Studio • Toon Boom Animation 29
Trango Broadband • Trango Broadband Wireless 47
WireTap Studio • Ambrosia Software Inc. 15

The index on this page is provided as a service to our readers.
The publisher does not assume any liability for errors or omissions.

THE MACTECH SPOTLIGHT

Austin Meyer

President, Laminar Research

What do you do?

I wrote X-Plane, and continue to upgrade it and administer the business now.

How long have you been doing what you do?

15 years, since college.

Are you Mac-only, or a multi-platform person?

I do all my daily operations, and maybe 99% of of my coding, on the Mac.

I just have to do the last 1% of the code or so on Windows to get the Windows version compiled and tested.

My market is maybe 50% Mac, 50% Windows so I DO need to maintain the Windows version of X-Plane... but I do 99% of the work on the Mac since C++ developed on one machine USUALLY works on the other

What's the coolest tech thing you've done using OS X?

Generated scenery for the entire planet earth from the terrain maps created by the space shuttle?

Flown virtually on mars using terrain data from the mars orbiting laser altimeter?

Flown virtually to orbit a larger airplane than has ever been built?

Where can we see a sample of your work?

www.x-plane.com

everybody loves new.

(we're right there with you.)

we know Macs

At Small Dog, we're obsessed with the new stuff too. We have been a part of the Mac community for over a decade, and we still get excited about it!

new products, new website

We sell and support Apple's newest innovations along with refurbished Macs, iPods and over 3,000 other Mac-friendly products. And we have a new website to make it easier to buy than ever.

Check out Smalldog.com to see what we have in store for you!

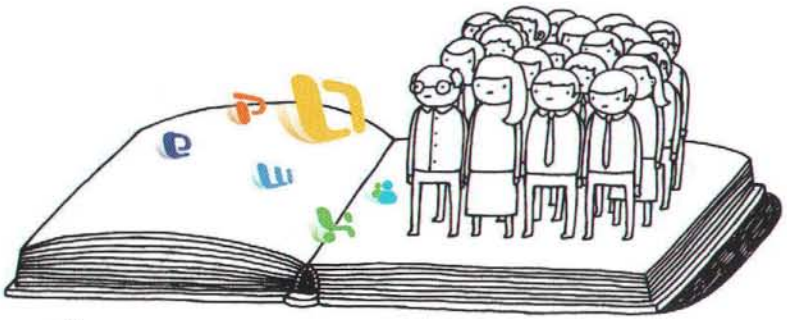
Small Dog Electronics

Always By Your Side

INDULGE! Save $15 off of your next purchase of $100 or more. Enter the coupon code "mactechnew" at checkout.

www.smalldog.com **800-511-MACS** Apple Specialist

Your potential. Our passion.®
Microsoft®



Everyone on the same page

Work together. Different machines? Different platforms? No matter. You can all speak the same language.

SimplifyYourWork2008.com

Office:mac 2008